Archive Backup System for OpenVMS

Installation Guide

Order Number: AA-QHD2S-TE

Software Version	Archive Backup System for OpenVMS Version V4.1
Required Operating System	OpenVMS VAX Version 6.2, 7.1, 7.2 and 7.3 OpenVMS Alpha Version 6.2, 7.1-2, 7.2-1, 7.2-2, 7.3 or 7.3-1
Required Software	Media, Device and Management Services for OpenVMS Version V4.1 DECnet (Phase IV) or DECnet-Plus(Phase V) TCP/IP Services for OpenVMS

January 2003

© 2003 Hewlett-Packard Development Company, L.P.

Confidential computer software. Valid license from HP and/or its subsidiaries required for possession, use, or copying.

Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Neither HP nor any of its subsidiaries shall be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for HP products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

Printed in the U.S.A.

Contents

Preface ix

1 Welcome To ABS

1.1 Installing ABS/MDMS 1-1
1.2 Updating ABS/MDMS 1-1
1.3 Upgrading from SLS to ABS/MDMS 1-1

2 Preinstallation

2.1 Decide Where to Install ABS/MDMS software 2-1
2.1.1 ABS/MDMS Server Software 2-1
2.1.2 ABS/MDMS Client Software 2-1
2.1.2.1 OpenVMS Clients 2-1
2.1.2.2 Windows Clients 2-2
2.1.2.3 Tru64 UNIX Clients 2-2
2.2 Verify Requirements 2-2
2.2.1 Required Privileges 2-2
2.2.2 Required OpenVMS Operating System Subclasses 2-2
2.2.3 Hardware, Software, and System Requirements 2-3
2.2.3.1 Required Hardware 2-3
2.2.3.2 Required Software 2-4
2.2.3.3 Optional Software 2-4
2.2.3.4 Required System Parameters 2-5
2.2.3.5 Required Process Account Quotas 2-5
2.2.3.6 Required Processes 2-6
2.2.3.7 Verify the Node is in the MDMS Database 2-8
2.3 Configure Hardware 2-8
2.4 Perform a System Backup 2-8
2.5 Registering ABS Licenses 2-9

3 Installing ABS Software

3.1 Installing Archive Backup System for OpenVMS Software 3-1
3.1.1 Installing ABS/MDMS Server Software 3-2
3.1.2 Installing ABS OpenVMS Client Software 3-4
3.1.3 Installing and Configuring ABS Windows Clients 3-4
3.1.4 Configuring ABS Tru64 UNIX Clients 3-6
3.1.4.1 Modifying the Appropriate Tru64 UNIX Files 3-7
3.1.4.2 Transferring the Tru64 UNIX Backup Agent Sources 3-8
3.1.4.3 Building the Tru64 UNIX Executables 3-9
3.1.5 Authorizing Windows and Tru64 UNIX Clients 3-12

4 Performing Postinstallation Tasks

4.1 Installing ABS for the First Time 4-2
4.2 Create a Node Object 4-2
4.3 Verifying ABS Installation 4-2
4.4 Providing Automatic Start Up and Shut Down 4-3
4.5 ABS and MDMS Startup Queues 4-3
4.6 Modifying ABS and MDMS Command Procedures 4-3
4.7 Removing SLS/MDMS V2.x Automatic Startup 4-4
4.8 Meeting OpenVMS Cluster Requirements 4-4
4.9 Configure MDMS 4-4
4.10 Convert ABS Policy Database to MDMS Database 4-5
4.11 Configure Remote Tape Drives 4-5
4.12 Granting the Appropriate ABS/MDMS Access Right Identifiers 4-6
4.12.1 Revoking An Access Rights Identifier 4-7
4.13 Verifying Windows and Tru64 UNIX Client Quotas 4-7
4.14 Adding Windows Parameter 4-7
4.15 Allowing ABS Access to All Files on the Windows Systems 4-8
4.16 ABS/MDMS Graphical User Interface (GUI) Installation 4-8
4.16.1 Installing the GUI on OpenVMS Alpha 4-9
4.16.2 Installing the GUI on Intel Windows NT/95/98 4-10
4.17 Running the GUI 4-10
4.17.1 Running the GUI on OpenVMS Alpha 4-10
4.17.1.1 Running the GUI on Windows NT/95/98 4-11
4.18 How to Remove ABS/MDMS to Version 4.0 Files 4-11
4.19 Configuration of ABS/MDMS for System Backup to Tape for Oracle Databases 4-11

A Examples of Authorizing Windows and Tru64 UNIX Clients

A.1 Adding Client Licenses A-1
A.2 Modifying Client Licenses A-1
A.3 Showing Client Licenses A-2
A.4 Removing Client Licenses A-3

B Upgrading ABS From the ABS-OMT License

C How to Delete ABS/MDMS From Your System

D Converting to ABS/MDMS V4.0

D.1 Introduction D-1
D.2 Converting SLS/MDMS V2.X Symbols and Database D-2
D.2.1 Executing the Conversion Command Procedure D-3
D.2.2 Resolving Conflicts During the Conversion D-3
D.2.3 Things to Look for After the Conversion D-5
D.2.4 Upgrading the Domain to MDMS V4 . . . . . D-10
D.2.5 Convert from MDMS Version 3 to a V2.X Volume Database D-10
D.3 SLS V2.x to ABS V4.0 Conversion Process D-11
D.3.1 Steps for Conversion D-11
D.3.1.1 Convert the MDMS Database D-11
D.3.1.2 Determine your use of SLS D-11
D.3.1.3 Converting SLS System Backups to ABS D-11
D.3.1.4 Converting User Backup policy D-17
D.3.1.5 Monitor ABS Activity D-17
D.3.1.6 Restoring from SLS History Sets D-17

D.3.2 Conversion Utility Reference D-18
D.3.2.1 Command Syntax D-18
D.3.2.2 Output Command File naming and contents D-18
D.3.3 SBK Symbols in ABS Terminology D-19
D.3.4 ABS Policy Attributes in SBK Terminology D-22
D.4 Converting ABS V2.x Catalogs to V4.0 Format D-26
D.5 Converting ABS V2.x/V3.x RDB Policy Database to ABS V4.0 (MDMS Server Database) D-26
D.6 Converting ABS V3.x RMS Policy Database to ABS V4.0 (MDMS Server Database) D-27

Tables

Table 2–1Required Software 2-4
Table 2–2Optional Software 2-4
Table 2–3System Parameters – Minimum Values 2-5
Table 2–4Required Installing Account Process Quotas 2-6
Table 2–5How to Start DECnet and the OpenVMS Queue Manager 2-7
Table 2–6How to Register Your ABS Licenses Using VMSLICENSE.COM 2-11
Table 3–1Stages of Installing ABS Software 3-1
Table 3–2How to Install ABS Software 3-3
Table 3–3Installing ABS OpenVMS Client Software 3-4
Table 3–4Stages of Installing an ABS Windows Client 3-5
Table 3–5Installing and Configuring an ABS Windows Client 3-5
Table 3–6Stages of Installing and Configuring an ABS Tru64 UNIX Client 3-6
Table 3–7Modifying the Appropriate Tru64 UNIX Files 3-7
Table 3–8Authorizing NT and Tru64 UNIX Client Nodes 3-13
Table 4–1Updating the DCL Tables 4-4

Preface

Intended Audience

This document is intended for storage administrators who are experienced OpenVMS system managers. This document should be used in conjunction with the *Introduction to OpenVMS System Management* manual.

Conventions

The following conventions are used in this document:

Convention	Description
{ }	In format command descriptions, braces indicate required elements.
[ ]	In format command descriptions, square brackets indicate optional elements of the command syntax. You can omit these elements if you wish to use the default responses.
boldface type	Boldface type in text indicates the first instance of a term defined in the Glossary or defined in text.
italic type	Italic type emphasizes important information, indicates variables, indicates complete titles of manuals, and indicates parameters for system information.
`Starting test ...`	This type font denotes system response, user input, and examples.
Ctrl/*x*	Hold down the key labels Ctrl (Control) and the specified key simultaneously (such as Ctrl/Z).
PF1 *x*	The key sequence PF1 *x* instructs you to press and release the PF1 key, and then press and release another key (indicated here by *x*).
n	A lowercase italic *n* denotes the generic use of a number. For example, 19*nn* indicates a four digit number in which the last two digits are unknown.
x	A lowercase *x* denotes the generic use of a letter. For example, *xxx* indicates any combination of three alphabetic characters.

Related Products

The following related products may be mentioned in this document:

Product	Description
HSM	HSM refers to Hierarchical Storage Management for OpenVMS software.
MDMS	MDMS refers to Media and Device Management Services for OpenVMS software.
OpenVMS	OpenVMS refers to OpenVMS operating system.
SLS	SLS refers to Storage Library System for OpenVMS software.

Associate Documents

The following documents are part of Archive Backup System for OpenVMS documentation set:

- *Archive Backup System for OpenVMS Installation Guide*
- *Archive Backup System for OpenVMS Guide to Operations*
- *Archive Backup System for OpenVMS MDMS Reference Guide*

1

Welcome To ABS

1.1 Installing ABS/MDMS

If you are installing Archive Backup System (ABS)/Media Device and Management System (MDMS) for the first time, read through this installation guide before installing the products. Also refer to the Archive Backup System Guide to Operations for information on configuring ABS/MDMS.

1.2 Updating ABS/MDMS

If you are updating ABS/MDMS, you should review the release notes provided in the installation kit. You may extract the release notes by entering the command:

```
@SYS$UPDATE:VMSINSTAL kit_name kit_location OPTIONS N
```

The upgrade from ABS/MDMS V3.x to V4.0 requires a conversion of the ABS policy database. See Appendix D for instructions on the conversion procedure.

Note

If you have ABS servers and clients which must be upgraded, they must all be done before running ABS V4.0. There is no rolling upgrade to this version. Upgrade the server and then upgrade each client before running any ABS saves/restores on the clients.

1.3 Upgrading from SLS to ABS/MDMS

If you are upgrading from SLS V.2.x to ABS/MDMS V4.0, see the information in Appendix D about converting SLS to MDMS, and converting SLS backup procedures to ABS.

Note

It is highly recommended that you do NOT run SLS in parallel with ABS V4.0. If you must use SLS in parallel with ABS/MDMS, be sure that you do NOT use the same media, drives, and jukeboxes in both products. This will cause inconsistencies in the databases for SLS and MDMS.

2

Preinstallation

To ready your OpenVMS system for either the ABS/MDMS server or ABS/MDMS client software installation procedure, you must perform certain tasks and requirements:

- Decide where to install the ABS/MDMS software
- Verify requirements
- Configure hardware
- Perform a system backup prior to the installation
- Install licenses

2.1 Decide Where to Install ABS/MDMS software

2.1.1 ABS/MDMS Server Software

Install ABS/MDMS server software on a disk with adequate space. This disk may be the system disk or another disk dedicated to ABS/MDMS. Disk space required for the installation are listed in Section 2.2.3.1. You should also take into account the space required for log files, catalogs and databases. Additional catalog disks may be added at a later time.

The ABS/MDMS server software should reside on a system which will be managing the policy and media databases for itself and any client nodes connected to it. The MDMS server provides access to the policy and media databases.

You must install ABS/MDMS server software on at least one OpenVMS node or OpenVMS cluster system.

There is one kit for both ABS/MDMS servers and clients. The installation will provide the appropriate configuration based on what server names are entered during the installation. The configuration may be modified at a later time by updating the node and server information in the MDMS database.

2.1.2 ABS/MDMS Client Software

2.1.2.1 OpenVMS Clients

ABS/MDMS client software should be installed on any OpenVMS node that can communicate with the ABS/MDMS server and for which you want to create ABS save/restore requests. This communication may be done using DECnet or TCP/IP.

Note

If you are upgrading, ABS V4.0 clients must be updated to V4.0 at the same time as the server. NO ROLLING UPGRADES are allowed. It is recommended that you shutdown the clients, upgrade the server, then upgrade each client.

2.1.2.2 Windows Clients

For Windows clients, you must install the ABS Windows software provided in the ABS/MDMS installation kit. You must also authorize access for Windows clients on the ABS server node. These tasks are described in Chapters 3 and 4.

2.1.2.3 Tru64 UNIX Clients

Tru64 UNIX clients do not require a separate client installation. However, you must transfer the GNU tar files provided by the ABS/MDMS installation kit to the client system, and then you must build the executable images on the True64 system. You must also authorize the clients on the ABS server node. These tasks are described in Chapters 3 and 4.

Both Windows and True64 UNIX clients have their backup operations occur on the ABS server node. The ABS server communicates with the Windows and Tru64 UNIX clients for data transfer and control.

Note

Given that you have the appropriate amount of licenses and adequate resources available, any number of ABS client nodes may be connected to a single ABS server node.

2.2 Verify Requirements

2.2.1 Required Privileges

To install ABS, log on to the SYSTEM account or to an account that has SETPRV or, at a minimum, has the following privileges enabled:

- CMKRNL
- WORLD
- SYSPRV
- TMPMBX
- NETMBX

Note that VMSINSTAL turns off the BYPASS privilege at the start of the installation procedure.

2.2.2 Required OpenVMS Operating System Subclasses

OpenVMS operating system comes with a variety of support options, or subclasses. Subclasses include such features as networking and RMS journaling. To use ABS, your system should have the following subclasses resident:

- Programming support
- Utilities
- System programming environment
- Secure user's environment
- Network support

How to verify:
For information about verifying these components, refer to either *OpenVMS VAX Installation Procedures* or *OpenVMS Alpha Installation Procedures*.

2.2.3 Hardware, Software, and System Requirements

To make sure that your system is ready for the installation, verify that your system meets the following requirements:

- Hardware
- Software
- System parameters
- Process account quotas
- Processes

2.2.3.1 Required Hardware

To install software, you must meet the following minimum hardware requirements:

- A VAX™ system or an Alpha system.
- Please refer to the OpenVMS documentation for minimum requirements of RAM memory for running OpenVMS. We recommend that you have at least 32 MB of memory on VAX systems and 128 MB of memory on Alpha systems.
- One or more tape drives if you plan to back up your data to removable media.
- One disk, such as a COMPAQ RD™ or RZ™ series disk.
- Adequate disk space. Verify that there are enough free blocks on the disk where you are installing the software.

Note

If you are providing remote drive support, you must answer *yes* to the remote drive question during MDMS installation procedure. This requires additional free disk space.

Enter the following command to show the amount of used disk space on your disk:

```
$ SHOW DEVICE disk-name
```

Disk space required during an full installation on an Alpha system is 214,500 blocks with 167,400 required after the installation. Disk space required on a VAX is 204,000 blocks with 120,000 required after the installation.

2.2.3.2 Required Software

Table 2-1 lists the software you must have installed on your system before you can install ABS.

Table 2–1 Required Software

Software	Purpose	System
DECnet Phase IV or DECnet Plus for OpenVMS or TCP/IP Services for OpenVMS	Provides network support **Note**: This software must be up and running *before* you start the installation procedure if you will be using a network with ABS/MDMS	ABS OpenVMS Server ABS OpenVMS Client
OpenVMS Operating System[a]	Provides OpenVMS and DCL capabilities.	ABS OpenVMS Server ABS OpenVMS Client

a. See *ABS Software Product Description (SPD)* for supported versions.

2.2.3.3 Optional Software

Table 2-2 describes the optional software you can use with ABS software

Table 2–2 Optional Software

Software	Purpose	System
ABS NT Client Software	Provides the ability to save data from Windows NT systems to ABS OpenVMS server system using ABS software (provided with ABS, not purchased separately).	Windows Client
C Compiler	Provides the ability to build the executable files on Tru64 UNIX clients	Tru64 UNIX Client
DCSC[a](Digital Cartridge Server Component)	If you have a StorageTek Automated Cartridge Server (ACS), you must install the DCSC software.	The system where the StorageTek silo resides.
TCP/IP Services for OpenVMS[a]	Provides network support for Windows and Tru64 UNIX clients.	ABS/MDMS OpenVMS Server,ABS/MDMS OpenVMS Clients, system where GUI will be run.
Media Robot Utility (MRU)	Provides library and loader testing, diagnostics, and control.	The OpenVMS system where the robotic device is physically connected
Java Run-Time Environment (JRE) V1.2 or 1.3.1	Provides run-time environment for MDMSView GUI.	The OpenVMS or Microsoft Windows system where the GUI will be run.

a. See *ABS Software Product Description (SPD)* for supported versions.

2.2.3.4 Required System Parameters

To install ABS, the system parameters must be set to the minimum value or higher. Table 2-3 lists the minimum system parameter values required for the installation procedure to run successfully. Depending on the kinds of programs and applications running at your site, you may need higher values.

Table 2–3 System Parameters – Minimum Values

System Parameter	Minimum Value	Dynamic
PQL_DENQLM	300	Y
GBLPAGES[a]	2000	N
GBLSECTIONS[a]	4	N
LOCKIDTBL	45000	N
PQL_MENQLM	300	Y
PQL_MPGFLQUOTA	10000	Y
PROCSECTCNT	100	N
PQL_MTQELM	200	Y

a. The values listed for these system parameters represent the number of free global pages and global sections required to install and run ABS, not the total number you need to operate your system and other software.

To see the current system parameter values on your system, enter the following command:

```
$ MCR SYSGEN
SYSGEN> SHOW/GEN
```

Result:
Shows the current values of all the system parameters. If you need to modify one or more of the system parameters, see the following example:

```
$ MCR SYSGEN
SYSGEN> SET GLBPAGES 2000
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```

The changed parameters should be added to SYS$SYSTEM:MODPARAMS.DAT for future changes made with AUTOGEN. You must then reboot the system so the non dynamic parameter values are recognized.

More information:
Refer to *OpenVMS System Manager's Manual: Managing System Parameters* for detailed information about required system parameters.

2.2.3.5 Required Process Account Quotas

The account you use to install ABS (typically the SYSTEM account) must have sufficient quotas to enable you to perform the installation. If your SYSTEM account quotas are the same as or higher than the default values provided with the OpenVMS operating system, then these values should be sufficient to install the software.

Table 2-4 summarizes the process quotas and the quotas that VMSINSTAL requires to perform the installation.

Table 2–4 Required Installing Account Process Quotas

Account Quota	Minimum Value
ASTLM	200
BIOLM	10000
BYTLM	18000
DIOLM	200
ENQLM	2048
FILLM	300
PGFLQUO	10000
TQELM	200

To see your current process quotas, see the following example:

```
$ MCR AUTHORIZE
UAF> SHOW SMITH
```

Result:
This command shows all your process quotas. If you need to increase your process account quotas, see the following example:

```
$ MCR SYS$SYTEM:AUTHORIZE
UAF> MODIFY SMITH/ENQLM=2048
UAF> EXIT
```

Note

If you are supporting NT or UNIX clients, to ensure successful save and restore operations, set the quotas to the following values from ABS OpenVMS server node:

```
UCX> SET PROTOCOL TCP/QUOTA=(SEND:50000,RECEIVE:50000)
```

If you have to reboot the machine, make sure that you reset these values after rebooting.

More information:
For detailed instructions about modifying account quotas, see the description of *Authorize Utility in OpenVMS System Management Subkit*.

2.2.3.6 Required Processes

Before beginning the installation procedure, check to see that DECnet Phase IV, DECnet Plus or TCP/IP and the OpenVMS Queue Manager are running. Network software is not required if you are on a standalone system and you will not be using the GUI. To see if these processes are active on your system, enter the following command:

```
$ SHOW SYSTEM
```

The following information is displayed for DECnet Phase IV:

```
OpenVMS V7.1 on node NODE1  8-AUG-1997 13:39:28.23Uptime 0 23:36:26
Pid        Process Name    State   Pri   I/O   CPU            Page flts Page
.
.
.
20A0022C   QUEUE_MANAGER   HIB     8     72    0 00:00:00.83    751       1210
.
.
.
20A00212   NETACP       HIB   10   285  0 00:00:02.84 338    666
```

The following information is displayed for DECnet Plus:

```
37C00215    NET$ACP         HIB   4   629  0 00:27:23.22   1894    2465
.
.
.
37C0024A    QUEUE_MANAGER   HIB   8  3333  0 00:07:45.24   1246    1766
```

The following information is displayed for UCX (earlier version of TCP/IP)

```
20A00444 UCX$INET_ACP HIB...
```

The following information is displayed for TCP/IP:

```
20A00555 TP_SERVER HIB...
```

If these processes are not active, follow the steps in Table 2-5.

Table 2–5 How to Start DECnet and the OpenVMS Queue Manager

Step	Action

1.	Start DECnet software.
	For DECnet Phase IV, enter the following command at DCL prompt:
	`$ @SYS$MANAGER:STARTNET.COM`
	For DECnet Plus, enter the following command at DCL prompt:
	`$ @SYS$STARTUP:NET$STARTUP.COM`

2.	Start OpenVMS Queue Manager. Enter the following command at DCL prompt: `$ START/QUEUE/MANAGER`

3.	Start the TCP/IP software.
	For TCP/IP Version 4-n, enter the following command at the DCL prompt:
	`$ @SYS$SYSTEM:UCX$STARTUP.COM`
	For TCP/IP Version 5.n, enter the following command at the DCL prompt:
	`$ @SYS$MANAGER:TCPIP$STARTUP.COM`

2.2.3.7 Verify the Node is in the MDMS Database

If this installation is not the **initial** installation of MDMS, you need to verify that the node you are installing MDMS on is in the MDMS database. Enter the following command on a node that has MDMS already installed on it and verify that the node you are installing MDMS on is in the database:

```
$ MDMS SHOW NODE node_name_you_are_installing_on
%MDMS-E-NOSUCHOBJECT, specified object does not exist
```

If the node is not in the database, you receive the %MDMS-E-NOSUCHOBJECT error message and you should create the node using the following command:

```
$ MDMS CREATE NODE node_name_you_are_installing_on
```

See the *Archive Backup System for OpenVMS MDMS Reference Guide* for the qualifiers to use.

If the node you are adding is an MDMS server node, the installation procedure will create the node using the /DATABASE qualifier. In addition, you need to edit all SYS$STARTUP:MDMS$SYSTARTUP.COM files in your domain and add this node to the definition of MDMS$DATABASE_SERVERS.

2.3 Configure Hardware

For your storage application to work, the hardware it depends on must be installed, connected, and configured to function with the operating system. To configure your hardware, see the hardware manuals provided with the hardware device. Also see the Archive Backup System Guide to Operations for troubleshooting suggestions.

If you are installing tape jukeboxes for use with ABS/MDMS, you should verify that the hardware is working correctly before installing ABS/MDMS. You may use the Media Robot Utility (MRU) to verify the hardware. If you do not have this software installed, it may be downloaded from

```
http://www.support.compaq.com/sms/mru/
```

Consider RDF Configuration

MDMS provides RDF software to facilitate operations that require access to remote, network connected tape drives. This allows you to copy data from a local site to a remote site, or copy data from a remote site to a local site. RDF requires that DECnet is installed and running.

RDF is not available if you are installing MDMS with the ABS-OMT license.

During the installation you will be asked questions on whether you want to install on this node, the software that will allow it to act as a server and/or client for the RDF software. You need to decide if you want the server and/or client installed on the node.

- Install the **RDF Server software** on all nodes that are connected to the tape drives used for remote operations.
- Install the **RDF Client software** on all nodes that initiate remote operations to tape drives on the RDF Server node.

2.4 Perform a System Backup

Compaq recommends that you perform a backup operation on the system disk before installing any software. For details about performing a backup operation on a system disk, see the *OpenVMS System Manager's Manual.*

2.5 Registering ABS Licenses

To use ABS software, you must register and load the licenses before you begin the installation procedure. This information is supplied in the License PAK document which is packaged along with *Archive Backup System for OpenVMS Cover Letter.*

To register a license under OpenVMS, use the following procedure:

1. Log in to the system where you will be installing the software. Log in under SYSTEM account, or enable your account with the privileges described in “Required Privileges” on page 2. Select one of the following methods to register the licenses:
   - At DCL prompt, enter the LICENSE REGISTER command with appropriate qualifiers that correspond to License PAK information. See .
   - Invoke SYS$UPDATE:VMSLICENSE.COM procedure. When it prompts you for information, respond with data from your License PAK. See Table 2–6.

If you plan to use ABS on more than one node in an OpenVMS Cluster, you must load the licenses on other nodes after you install ABS. See Table 2-6, Step 10 for instructions.How to Register Your ABS Licenses Using the LICENSE REGISTER Command

Step	Action
Step 1.	Enter the LICENSE REGISTER command with the product name followed by a dash (-): **$** LICENSE REGISTER ABS-SERVER-VAX - ! Register this license on the ABS VAX server node **$** LICENSE REGISTER ABS-SERVER-ALPHA - ! Register this license on the ABS Alpha server node **$** LICENSE REGISTER ABS-CLIENT-VAX - ! Register this license on all ABS VAX client nodes **$** LICENSE REGISTER ABS-CLIENT-ALPHA - ! Register this license on all ABS Alpha client nodes **$** LICENSE REGISTER ABS-NT-CLIENT-USER - ! Register this license on the ABS server node where you plan to support NT clients. **$** LICENSE REGISTER PAB-UNIX-CLIENT-USER - ! Register this license on the ABS server node where you plan to support UNIX clients. **$** LICENSE REGISTER ABS-OMT - ! Register this license on all ABS server nodes where you have to install the ABS software **$** LICENSE REGISTER ABS-OMT-UPG - ! Register this license on all ABS server nodes where you want to upgrade the ABS-OMT license to the full ABS product **Important**: Enter a dash (-) at the end of each command from Steps 1 through 8.
Step 2.	Enter the /ISSUER qualifier information, assigning the value DEC between quotation marks. _$ /ISSUER=”DEC” -
Step 3.	Enter the /AUTHORIZATION qualifier information, assigning it the value from the AUTHORIZATION NUMBER[a] entry of the PAK: _$ /AUTHORIZATION=xxxxxx -

Step	Action
Step 4.	Enter the /PRODUCER qualifier information, assigning the value DEC in quotes: _$ /PRODUCER=”DEC” -
Step 5.	Enter the /UNITS qualifier information, assigning it the value from the UNITS[a] entry of the PAK _$ /UNITS=nn -
Step 6.	Enter the /DATE qualifier information, assigning the product’s release date value from the PRODUCT RELEASE DATE[a] entry of the PAK: _$ /DATE=dd-mmm-yyyy -
Step 7.	Enter the /AVAILABILITY qualifier information, assigning the value from the AVAILABILITY TABLE CODE[a] entry of the PAK: _$ /AVAILABILITY=x -
Step 8.	Enter the /OPTIONS qualifier information, assigning the value from the KEY OPTIONS[a] entry of the PAK: _$ /OPTIONS=xxxxxx -
Step 9.	Enter the /CHECKSUM qualifier information, assigning the value from the CH[a] entry of the PAK: `_$ /CHECKSUM=1-xxxx-xxxx-xxxx-xxxx` **Important:** Do NOT end the entry with a dash.
Step 10.	Invoke the LICENSE LOAD command with the product name: **$** LICENSE LOAD product_name

a. This information is supplied in the LICENSE PAK document (provided with the software kit).

Table 2–6 describes how to register your license using the command procedure.

Table 2–6 How to Register Your ABS Licenses Using VMSLICENSE.COM

Step	Action
Step 1.	From the system prompt, enter the following command: `$ @SYS$UPDATE:VMSLICENSE.COM`
Step 2.	Select Option 1. "REGISTER a Product Authorization Key"

Step 3.	Answer the questions according to the information supplied in the LICENSE PAK document (provided with the software kit).
	
	The following is only an example. Supply the information provided in the PAK to the prompts:
	
	Type ? at any prompt for a description of the information requested. Press Ctrl/Z at any prompt to return to the main menu.
	
	`Issuer[DEC]`
	`Authorization Number[]:`
	`Authorization Number[]:ALS-NQ-1996JUN10-181`
	`Product Name[]:ABS-SERVER`
	`Producer[DEC]:`
	`Number of Units[]:1050`
	`Version[]:`
	`Product Release Date[]:`
	`Key Termination Date[]:`
	`Availability Table Code[]:H`
	`Activity Table Code[]:`
	`Key Options[]:MOD_UNITS,ALPHA`
	`Product Token[]:`
	`Hardware-Id[]:`
	`Checksum[]:2-PIBA-KIPP=BIGE-DDHC`

Step 4.	Verify the information that you entered is correct. Enter **Yes**.
Step 5.	To exit the command procedure, select Option 99.

3

Installing ABS Software

This chapter contains instructions for installing Archive Backup System for OpenVMS software.

Before proceeding with the installation procedure, make sure you have completed all of the following preinstallation tasks:

- Did you decide where to install ABS server and client software?
- Did you set your default directory to SYS$UPDATE?
- Did you log into an account with the proper quotas and privileges?
- Did you perform a system backup operation?
- If you are doing an upgrade installation, did you shutdown ABS and MDMS?
- Did you verify the hardware and disk space requirements?
- Did you verify the software requirements?
- Did you check to see if DECnet (Phase IV), DECnet-Plus or TCP/IP, and the QueueManager are running?
- Did you register the appropriate licenses?

3.1 Installing Archive Backup System for OpenVMS Software

See Table 3–1 for the stages of installing and configuring ABS/MDMS software.

Note

Before installing ABS in a real time, storage management environment, COMPAQ recommends that you first install and configure ABS in a test environment. If you are not satisfied with the test installation, delete ABS and reinstall it.

Table 3–1 Stages of Installing ABS Software

Stage	Action
1.	Install ABS server software as described in Section 3.1.1. **Note**: If you are installing ABS in a mixed architecture environment (VAX and Alpha systems resident in a single OpenVMS Cluster), install the software in a common location. This will place the database and catalog files in a common location for the entire cluster. Images will be placed on the system disks for each system. Install ABS/MDMS on each node.
2.	Install ABS OpenVMS client software as described in Section 3.1.2. **Note**: OpenVMS client support is not available with an ABS-OMT license.

Table 3–1 Stages of Installing ABS Software

3.	Install ABS Windows client software as described in Section 3.1.3.
4.	Install ABS Tru64 UNIX clients as described in Section 3.1.4.
5.	Authorize Windows and Tru64 UNIX clients as described in Section 3.1.5.
6.	Follow the Post-Installation steps in Chapter 4.

3.1.1 Installing ABS/MDMS Server Software

ABS/MDMS installation procedure consists of a series of questions and informational messages. The procedure will give you an option of doing a standard installation or not. The standard installation will install the ABS/MDMS software in the following manner:

```
ABS will be installed with the following options:
Device for ABS/MDMS - SYS$COMMON (will use current values if either MDMS or
ABS are already installed)
UIC for the ABS account - [311,311]
Nodes for the database severs - Current node or already existing server nodes
What scheduling option do you want - INTERNAL
Are you using backup-via-shelving - YES (If HSM is present)

MDMS will be installed with the following options:
Device for ABS/MDMS - SYS$COMMON (or current installed location)
UIC for MDMS Account - [312,312] (or will be constructed from ABS UIC)
Node for database servers - Current node or already existing server nodes
Do you want to replace MRD$RTL.EXE - YES

MDMS GUI will files will be installed with the following options:
Do you want the MDMS GUI installed on Alpha OpenVMS - YES (if on Alpha)
Do you want files extracted for Microsoft Windows on Intel - YES

Remote Device Facility (RDF) will be installed with the following options:
Do you want MDMS support for RDF Server - YES
Do you want MDMS support for RDF Client - YES

Questions always asked:
Do you want to purge files - YES
Do you want to run the IVP - YES (except if ABS-OMT installation)
```

If for any reason you need to abort the installation procedure, at any time you can press CTRL/Y and the installation procedure deletes all files it has created up to that point and then exits. From this point, you can restart the installation procedure again.

Follow the steps in Table 3–2 to install ABS software.

Table 3–2 How to Install ABS Software

Step	Action

1.	**Invoke VMSINSTAL**:
	
	`$ @SYS$UPDATE:VMSINSTAL saveset-name drive-name OPTIONS N`
	
	To start the installation, invoke the VMSINSTAL command procedure from a privileged account, such as the SYSTEM account. VMSINSTAL is in the SYS$UPDATE directory. The following list defines the elements of the VMSINSTAL command procedure:
	
	save set name
	
	The installation name for the component.
	
	`ABS041 (for example)`
	
	drive-name
	
	The name of the drive where the media that contains the kit is located. For example, MTA0: is the device name for a tape drive or device:[directory] can be the CD-ROM drive name. It is not necessary to use the console drive for this installation. However, if you do use the console drive, you should replace any media you removed once the installation is complete.
	
	OPTIONS N
	
	An optional parameter that indicates you want to see the question on release notes. If you do not include the OPTIONS N parameter, VMSINSTAL does not ask you about the release notes. You should review the release notes before proceeding with the installation in case they contain additional information about the installation.
	
	Note:
	If you are restarting the installation and have already reviewed the release notes, you do not need to specify OPTIONS N. If you specify more than one option, separate them with commas (OPTIONS A,N).
	
	The following examples invoke VMSINSTAL to install ABS from the tape drive MTA0: and shows the responses. This example uses the OPTIONS N release note parameter.
	
	`$ @SYS$UPDATE:VMSINSTAL saveset_name MTA0: OPTIONS N`
	
	`OpenVMS VAX Software Product Installation Procedure V7.2`
	`It is 21-JUL-1996 at 10:00`
	`Enter a question mark (?) at any time for help.`
	
	If you do not supply either the product name or the drive name, VMSINSTAL prompts you for this information later in the installation procedure.
	
	Note:
	VMSINSTAL does not prompt you for any options, so be sure to include OPTIONS N on the VMSINSTAL command line to access the release notes during the installation procedure. See OpenVMS documentation located in the OpenVMS System Management Subkit for detailed information on these options.

3.1.2 Installing ABS OpenVMS Client Software

Note

ABS/MDMS V4.0 does not allow a rolling upgrade on clients. You must install V4.0 on all of your clients before using the V4.0 ABS/MDMS server.

Note

OpenVMS client support is not available with an ABS-OMT license.

When installing ABS software, notice that ABS does not provide two separate software kits. Instead, installation of ABS OpenVMS server or client software is determined by the OpenVMS node name that you enter during the installation procedure.

Table 3–3 describes how to install and configure an ABS OpenVMS client.

Note

In Table 3–3, ABS server node is referred to as SVNODE::, and ABS client node is referred to as CLNODE::

Table 3–3 Installing ABS OpenVMS Client Software

Step	Action
1.	Install ABS software on the OpenVMS client node as on the server node, with one exception. When the installation procedure prompts for the node name for the server (ABS OpenVMS server node), do not accept the default node name (client node name). Instead, enter the OpenVMS Cluster alias or OpenVMS node name that you entered when you installed ABS OpenVMS server software: `* Node name list for Database Servers : SVNODE` **Result**: ABS installs only the client portion of ABS software on the node named CLNODE::.
2.	Create save and restore requests for OpenVMS clients as described in *Archive Backup System for OpenVMS Guide to Operations*.
3.	Create (or modify) storage and environment policies. *Archive Backup System for OpenVMS Guide to Operations* describes how to create those policies.
4.	Create system and user backup operations using the correct storage and environment policies. *Archive Backup System for OpenVMS Guide to Operations*, provides instructions for these tasks.

3.1.3 Installing and Configuring ABS Windows Clients

Requirements:

If you use FTP to copy the setup files to the Windows System, be sure to copy the files in binary mode.

Table 3–4 describes the stages of installing and configuring an ABS Windows client node.

Table 3–4 Stages of Installing an ABS Windows Client

Stage	Action
1.	Register ABS Windows license on ABS OpenVMS server node. See Section 2.5 for instructions.
2.	Install ABS Windows client software on Windows client system as described in Table 3–5. **Note:** You must perform a separate installation on each Wndows client node that you want ABS to back up.
3.	Authorize the Windows client systems that you plan to back up using ABS (described in Section 3.1.5 and *Appendix A*.
4.	Create save and restore requests for the Windows client system by using the GUI on the Windows client system, or by using the GUI or DCL on ABS OpenVMS server node. See *Archive Backup System for OpenVMS Guide to Operations* and *Archive Backup System For OpenVMS MDMS Reference Guide* for instructions about creating save and restore requests for Windows clients.

To install and configure the Windows client software, use the procedure in Table 3–5.

Table 3–5 Installing and Configuring an ABS Windows Client

Step	Action
1.	Copy all files from one of the following directory located on ABS OpenVMS server node to the Windows client system where you plan to install the Windows client software, or to a location accessible by the Windows client system. The Windows client software is provided during ABS server installation procedure. ABS creates the appropriate directory and places the Windows client software kit in the following directory: `ABS$ROOT:[CLIENTS.NT.INTEL]`

Table 3–5 Installing and Configuring an ABS Windows Client

Step	Action
2.	Run SETUP.EXE from this location to install ABS Windows client software. **Result:** ABS Windows client installation procedure prompts for information about the ABS server and the local host port number. Answer the prompts exactly as you answered them during ABS server installation procedure: **ABS server node**–Enter the node(s) on which ABS server software has been installed and will be providing ABS services for the Windows client backup operations. If multiple nodes are specified, separate them with semi-colons. This node also verifies connection requests. If a server node is multihomed (has more than one IP address/name), specify all the names which may be used during a backup request. **Local host port number**–The Windows client system uses a TCP/IP port to communicate with ABS server to initiate save and restore requests. The default port number is 1800. If you decide to change the port number, it is limited to a range between 1024 and 65535. This port number is arbitrary, but it must match the port number you use when you authorize Windows clients, described in Appendix A. To test to see if port number 1800 is already in use, enter the following command from ABS OpenVMS server system prompt: `$ TELNET node 1800` Where: *node* is the node name of the OpenVMS server node **Note**: Make sure that the number you provide does not conflict with a previously installed software application.
3.	Authorize the Windows clients you plan to back up using ABS as described in Section 3.1.5 and Appendix A.

3.1.4 Configuring ABS Tru64 UNIX Clients

To allow ABS to perform backup and restore operations for Tru64 UNIX clients, you must configure access between the OpenVMS systems that run ABS server software and the Tru64 UNIX client systems that contain the files to be backed up. The stages of installing and configuring a Tru64 UNIX client is described in Table 3–6.

Note

There is no extra software kit provided for Tru64 UNIX clients.

Table 3–6 Stages of Installing and Configuring an ABS Tru64 UNIX Client

Stage	Action
1.	Register ABS Tru64 UNIX license on the ABS OpenVMS server node. See Section 2.11 for instructions.

Table 3–6 Stages of Installing and Configuring an ABS Tru64 UNIX Client

Stage	Action
2.	Register ABS Tru64 UNIX license on the ABS OpenVMS server node. See Section 2.11 for instructions.
3.	Modify the appropriate files on the Tru64 UNIX client system as described in the Section 3.1.4.1.
4.	Transfer the gtar and gzip sources from ABS OpenVMS server node to each Tru64 UNIX client system that you intend to back up using ABS. See Section 3.1.4.2.
5.	Build the executables on each Tru64 UNIX client system that you plan to back up using ABS as described in Section 3.1.4.3.
6.	Authorize the Tru64 UNIX clients as described in Section 3.1.5 and Appendix A.
7.	Create save and restore requests for theTru64 UNIX client from the OpenVMS server node. See *Archive Backup System for OpenVMS Guide to Operations* and *Archive Backup System for OpenVMS MDMS Reference Guide* for instructions about creating save and restore requests for Tru64 UNIX clients.

3.1.4.1 Modifying the Appropriate Tru64 UNIX Files

Table 3–7, lists the files that you need to modify on each Tru64 UNIX client system that ABS is going to back up, and it describes the modifications to make for those specific files.

Note

Tru64 UNIX is a case sensitive system. Be sure to enter the commands on the Tru64 UNIX client system exactly as they are shown in Table 3–7 and Example 3–1.

Table 3–7 Modifying the Appropriate Tru64 UNIX Files

File	Modification

/.rhosts [a]	Replace the ASCII readable internet address with ABS OpenVMS server nodes. The file format is:
	`# readable ip address  account`
	`node01.vms.real.node  ABS  #ABS on node NODE01`
	In this example, replace node01.vms.real.node with ABS OpenVMS server node names. The account name must stay the same (ABS), and it *must be specified in capital letters*.
	Example of /.rhosts:
	`node01.vms.real.node  ABS`
	`node02.vms.real.node  ABS`
	Requirement:
	You must always modify the /.rhosts file. If the file does not exist, then you must create it. Be sure that the /.rhosts file is located in the root directory and that is owned by the root account because ABS uses this directory during the backup operation.

Table 3–7 Modifying the Appropriate Tru64 UNIX Files

File	Modification

/etc/hosts[a]	List the numeric internet address and the ASCII readable internet address of ABS OpenVMS server nodes. The file format is:
	
	`# Internet Address  Hostname  # Comments`
	`nn.nn.nn.nn  node01.vms.real.node  # example entry for hosts file`
	
	Where:
	
	• *nn.nn.nn.nn* is the numeric internet address of ABS node that executes save and restore requests.
	
	• *node01.vms.real.nod*e is the ASCII readable internet address of nn.nn.nn.nn. You must enter every node name on which ABS may execute.
	
	Example of /etc/hosts:
	
	`01.02.03.04 node01.vms.real.node  # node01 running ABS`
	`01.02.01.04 node02.vms.real.node  # node02 running ABS`
	
	Note:
	If ABS OpenVMS server nodes are already listed in the file /etc/hosts, you do not need to add them again.

a. File is not replaced, only modified.

3.1.4.2 Transferring the Tru64 UNIX Backup Agent Sources

During the installation of ABS server software, the installation procedure creates a directory named ABS$ROOT:[CLIENTS.UNIX] on ABS server node. This directory contains the following two uncompressed sources that make up the Tru64 UNIX backup agent:

- ABS$ROOT:[CLIENTS.UNIX]TAR-1_12.TAR
- ABS$ROOT:[CLIENTS.UNIX]GZIP-1_2_4.TAR

To configure a Tru64 UNIX client, you must transfer the gtar and gzip sources to each Tru64 UNIX client system that ABS is going to back up, build the executables, and place them in /usr/bin.

Refer to Example 3–1 to transfer the gtar and gzip sources from ABS server node to theTru64 UNIX client system.

Example 3–1 Transferring the Backup Agent Sources

```
u_node> ftp node01    # Connect to the ABS OpenVMS Server Node
Connected to node01.vms.dec.com.
220 node01 FTP Server (Version 3.3) Ready.
Name (node01:user1): user1
331 Username USER1 requires a Password.
Password:
230 User logged in.
Remote system type is VMS.

ftp> cd abs$root:[clients.unix] # Change to the directory that contains the
files
250 CWD command successful.
ftp>  pwd
257 "ABS$ROOT:[CLIENTS.UNIX]" is current directory.
ftp> ls  # List the files in this directory
```

```
200 PORT command successful.
150 Opening data connection for  (16.82.16.75,1174)
gnu_general_public_license.txt;4
gnu_readme_where_to_get.txt;4
gzip-1_2_4.tar;4
tar-1_12.tar;4
226 NLST Directory transfer complete.

ftp> bin   # set the file transfer mode to binary
200 TYPE set to IMAGE.

ftp> get   # Get the sources
(remote-file) tar-1_12.tar
(local-file) tar-1_12.tar
200 PORT command successful.
150 Opening data connection for tar-1_11_8.tar (16.82.16.75,1178)
226 Transfer complete.
2662400 bytes received in 5.7 seconds (4.6e+02 Kbytes/s)

ftp> get  # Get the sources
(remote-file) gzip-1_2_4.tar
(local-file) gzip-1_2_4.tar
200 PORT command successful.
150 Opening data connection for gzip-1_2_4.tar (16.82.16.75,1494)
226 Transfer complete.
798720 bytes received in 1.8 seconds (4.3e+02 Kbytes/s)

ftp> quit
221 Goodbye
```

3.1.4.3 Building the Tru64 UNIX Executables

After you have transferred the gtar and gzip sources, you are required to build the executables on the Tru64 UNIX client system. With a Tru64 UNIX OS version upgrade, it is **mandatory** to rebuild ABSgtar and ABSgzip executables on the Tru64 UNIX client.

The following sections describe how to build the tar and gzip executables.

3.1.4.3.1 Building the tar Executable –Use the following procedure to build the tar executable:

1. Use native tar to expand the tar file
2. Change directory to the tar directory
3. Enter the command ./configure --disable-nls
4. Enter the command make to build the tar image
5. Verify the tar image was created
6. Change directory to src
7. Verify that it is an executable image
8. Change to super user
9. Copy the executable from src/tar to usr/bin/ABSgtar
10. Change the protection on the image
11. Display the complete directory
12. Exit super user
13. Perform a cleanup operation

Example of tar

```
u_node> tar -xvf tar-1_12.tar [1]
tar-1.12/README
tar-1.12/AUTHORS
     .

     .

     .
tar-1.12/po/sv.gmo

u_node> cd tar-1.12 [2]
u_node> configure --disable-nls [3]
creating cache ./config.cache
checking host system type... alpha-dec-osf3.2

        .

        .

        .
creating config.h

u_node> make [4]
for subdir in doc lib intl src scripts po; do \

echo making all in $subdir; \
(cd $subdir && make CC='gcc' CFLAGS='-g -O' LDFLAGS='' LIBS=''
prefix='/usr/local' exec_prefix='/usr/local'
bindir='/usr/local/bin' libexecdir='/usr/local/libexec'
infodir='/usr/local/info' infodir='/usr/local/info'
libexecdir='/usr/local/libexec' all) || exit 1; \

done
making all in doc

        .

        .

        .
make[1]: Leaving directory `/usr/users/user1/tar-1.11.8/po'

u_node> ls src 5
Makefile   checktar.sh  extract.o  list.c   open3.h  rmt.o    tar.h
        .

        .

          .
buffer.c   diffarch.o gnu.c    names.c  rmt.c    tar [5]

u_node> cd src  [6]
u-node> file tar [7]
tar:  COFF format alpha dynamically linked, demand paged
     executable or object module not stripped
- version 3.11-8

u-node> su [8]
Password:
```

```
# cp tar /usr/bin/ABSgtar [9]
# chmod ugo+x /usr/bin/ABSgtar [10]
# ls -l /usr/bin/ABSgtar [11]
-rwxr-xr-x  1 root system 655794 Jan 24 11:07 ABSgtar
# exit [12]

u-node> cd .. [13]
u-node> rm -rf tar-1_12 [13]
%rm -f tar-1_12.tar [13]
```

3.1.4.3.2 Building the gzip Executable – Use the following procedure to build the gzip executable:

1. Use the native tar to expand the gzip-1_2_4.tar image
2. Change directory to gzip-1.2.4
3. Enter the command ./configure
4. Enter the command make
5. Verify that the gzip file is there
6. Make sure it is an executable
7. Change to super user
8. Copy gzip file to /usr/bin/ABSgzip
9. Change the protection on the image
10. Exit super user
11. Perform a cleanup operation

Example of gzip

```
u_node> tar -xvf gzip-1_2_4.tar [1]
gzip-1.2.4/README

        .

        .

        .
gzip-1.2.4/primos/include/sysTypes.h

u_node> cd gzip-1.2.4 [2]

u_node> ./configure [3]
checking for gcc

        .

        .

        .
checking for gzip to derive installation directory prefix
chose installation directory prefix /usr/local
creating config.status
creating Makefile
```

```
u_node> make [4]
gcc -c -DSTDC_HEADERS=1 -DHAVE_UNISTD_H=1 -DDIRENT=1 -O gzip.c

        .

        .

        .
gcc -O -o gzip gzip.o zip.o deflate.o trees.o bits.o unzip.o
inflate.o util.o crypt.o lzw.o unlzw.o unpack.o unlzh.o getopt.o
/usr/ucb/ld:
Warning: Linking some objects which contain exception information sections
and some which do not. This may cause fatal runtime exception handling prob-
lems (last obj encountered without exceptions was crypt.o).

rm -f gunzip zcat
ln gzip gunzip
ln gzip zcat

u_node> ls gzip [5]
gzip
u_node> file gzip [6]
gzip: COFF format alpha dynamically linked, demand paged executable or object
module not stripped
- version 3.11-8

u-node> su [7]
Password:
# cp gzip /usr/bin/ABSgzip [8]
# chmod ugo+x /usr/bin/ABSgzip [9]
# ls -l /usr/bin/ABSgzip
-rwxr-xr-x  1 root  system 654785 Jan 24 11:08 ABSgzip
# ln -s /usr/bin/ABSgzip /usr/bin/gzip
# exit [10]

u-node> cd .. [11]
u-node> rm -rf gzip-1.2.4 [11]
u-node> rm -f gzip-1_2_4.tar [11]
```

Note

If you have problems transferring or building the tar or gzip files, see your Tru64 UNIX system manager.

3.1.5 Authorizing Windows and Tru64 UNIX Clients

After you have registered and loaded the Windows or Tru64 UNIX client license on ABS server node, run the authorization executable file to authorize the Windows or Tru64 UNIX client node names that you intend to back up using ABS. You must authorize access on the ABS OpenVMS server system for each Tru64 UNIX and Windows system that you are going to back up using ABS.

Table 3–8 describes how to authorize Windows and Tru64 UNIX client nodes.

Note

ABS Windows and Tru64 UNIX client licenses are sold in units according to the number of nodes that you want to support: 1, 5, 10, 25, 50, or 100. ABS calculates the number of nodes authorized versus the number of units the license allows. You cannot authorize more Windows or Tru64 UNIX clients than the number of units allowed by the Windows or Tru64 UNIX client license that you have purchased.

Table 3–8 Authorizing NT and Tru64 UNIX Client Nodes

Step	Action
1.	Enter the following command from ABS server: `$ RUN SYS$SYSTEM:ABS$CLIENT_LICENSE.EXE`

2.	Add the node names of the Tru64 UNIX or Windows client nodes. When prompted, specify an NT or UNIX client:
	`Would you like to Add/Modify/Remove/Show the Client License?: ADD`
	`Enter Node Name: CLIENT_NODE_NAME`
	`Client Node Type (UNIX or NT) [UNIX]: NT`
	`Enter TCPIP Port Number [1800]:`
	Note:
	The port number is arbitrary, but it must match the port number you use when you install the NT clients.

3.	If the client node is an Windows client, enter the port number. The default is 1800.
4.	Make sure that the logical named ABS$CLIENT_DB is defined as /SYSTEM/EXEC and that it translates to ABS$ROOT:[DATABASE]CLIENT_DB.DAT. This logical name should be defined during the startup procedure. You can verify that the logical name is defined correctly by entering the following command: `$ SHOW LOGICAL/FULL ABS$CLIENT_DB`

More Information:
For an example of adding, modifying, removing and showing Windows or Tru64 UNIX clients, see *Appendix A*.

4

Performing Postinstallation Tasks

Complete ABS/MDMS postinstallation tasks described in this chapter after you have successfully installed ABS/MDMS OpenVMS server or client software:

- Section 4.1 describes how to initialize the ABS database if this failed during the installation.
- Section 4.2 describes how to create a node object for MDMS.
- Section 4.3 describes how to verify the installation was successful.
- Section 4.4 explains how to edit the startup and shutdown files.
- Section 4.5explains the batch queues used by ABS and MDMS startup procedures.
- Section 4.6 explains editing the ABS and MDMS command procedures.
- Section 4.7 describes removing the SLS/MDMS V2.x Automatic Startup.
- Section 4.8 describes the requirements for an OpenVMS Cluster installation.
- Section 4.9 describes configuring MDMS.
- Section 4.10 describes converting the ABS Policy Database to the MDMS database.
- Section 4.11 describes configuring remote tape drives.
- Section 4.12 explains how to set up access right identifiers.
- Section 4.13 explains how to set the quotas if you are supporting Tru64 UNIX and Windows clients.
- Section 4.14 describes the need of adding Windows parameter, before issuing multivolume SAVE requests.
- Section 4.15 describes the procedure to allow ABS to access all files on Windows systems.
- Section 4.16 describes the MDMS graphical user interface (GUI) installation.
- Section 4.17 describes how to run the graphical user interface (GUI).
- Section 4.18 describes how to remove ABS/MDMS V4.0 files if required.
- Section 4.19 describes where to find information on configuring ABS/MDMS for System Backup to Tape for Oracle Databases.

Note

OpenVMS client support is not available with an ABS-OMT license.

4.1 Installing ABS for the First Time

If you are installing ABS as a new installation, database initialization programs may fail to run. To initialize the database with the default storage policies and execution policies, run the following executable:

```
RUN SYS$SYSTEM:ABS$DB_INIT.EXE
```

4.2 Create a Node Object

If this is the initial installation of MDMS, you may need to create the node object in the MDMS node database for this node. Use the MDMS CREATE NODE command to create this initial database node. Refer to the command reference guide for the qualifiers for this command. The following is an example:

```
$ MDMS CREATE NODE NABORS -
! NABORS is the DECnet Phase IV node name or a
! name you make up if you do not use DECnet
! Phase IV in your network
/DATABASE_SERVER -
! a potential database node
! must also be defined in
! in SYS$STARTUP:MDMS$SYSTARTUP.COM
/TCPIP_FULLNAME=NABORS.SITE.INC.COM -
! the TCP/IP full node name if you
! are using TCP/IP you need this if
! you are using the GUI
/DECNET_FULLNAME=INC:.SITE.NABORS -
! this is the full DECnet Phase V node name
! do not define if you do not have DECnet Phase V on this node
! be sure to define if you have DECnet Phase V installed on this node
/TRANSPORT=(DECNET,TCPIP)
! describes the transports that listeners are
! started up on
```

4.3 Verifying ABS Installation

If you did not execute the IVP during the installation procedure, you can execute it immediately after installing ABS/MDMS software. Enter the following command at the DCL system prompt:

```
$ @SYS$TEST:ABS$IVP.COM

$ @SYS$TEST:MDMS$IVP.COM
```

Note

Support for installation IVP procedure is not available with an ABS-OMT license.

If an error occurs during the IVP, the following message is displayed:

```
ABS Version V4.1 Installation Verification Procedure failed.

%VMSINSTAL-E-IVPFAIL, The IVP for ABS Version V4.1 has failed.
```

Errors can occur during the installation if any of the following conditions exist:

- ABS is currently running
- The operating system version is incorrect
- A prerequisite software version is incorrect
- Quotas necessary for successful installation are insufficient

- System parameter values for successful installation are insufficient
- The OpenVMS help library is currently in use
- The product license has not been registered and loaded

For descriptions of the error messages generated by these conditions, see the OpenVMS documentation on system messages, recovery procedures, and OpenVMS software installation. If you are notified that any of these conditions exist, you should take the appropriate action as described in the message.

4.4 Providing Automatic Start Up and Shut Down

You must edit the startup and shutdown files to provide automatic startup and shutdown of ABS/MDMS software. To make sure that ABS/MDMS automatically starts up and shuts down, follow these steps:

Step 1. Add the following command line to the system startup file named SYS$MANAGER:SYSTARTUP_ VMS.COM:

```
$ @SYS$STARTUP:ABS$STARTUP
```

Step 2. Add the following line to the system shutdown file named SYS$MANAGER:SYSHUTDWN.COM:

```
$ @SYS$STARTUP:ABS$SHUTDOWN
$ @SYS$STARTUP:MDMS$SHUTDOWN
```

Step 3. When using MDMS with ABS, the MDMS startup is executed automatically. A logical name is defined by the MDMS startup which is needed by ABS.

4.5 ABS and MDMS Startup Queues

ABS$STARTUP uses the ABS$<node> queue to start the ABS processes. This queue will always exist after an ABS installation.

MDMS$STARTUP uses one of several queues to startup the MDMS$SERVER process. There is a logical name, MDMS$STARTUP_QUEUE, which may be defined in MDMS$SYSTARTUP.COM. This logical points to the queue which MDMS$STARTUP.COM will use. If you do not have this defined, it will attempt to use the ABS$<node> queue, then SYS$BATCH. If the logical is not defined and neither ABS$node or SYS$BATCH exist, the MDMS startup will fail.

4.6 Modifying ABS and MDMS Command Procedures

If you are updating ABS/MDMS and you have made customizations to any of the ABS/MDMS command procedures, you may need to modify those procedures to work with ABS/MDMS V4.0. There are four procedures for which we provide template files. If the template files have changed, you should include your customizations in the new template files and then rename them to the command procedure name. The four procedures and the templates are:

```
SYS$STARTUP:ABS$SYSTARTUP.COM - SYS$STARTUP:ABS$SYSTARTUP.TEMPLATE

SYS$STARTUP:MDMS$SYSTARTUP.COM - SYS$STARTUP:MDMS$SYSTARTUP.TEMPLATE

MDMS$SYSTEM:MDMS$EXT_QUEUE_MANAGER.COM - MDMS$SYS-
TEM:MDMS$EXT_QUEUE_MANAGER.TEMPLATE

MDMS$SYSTEM:MDMS$EXT_SCHEDULER.COM - MDMS$SYSTEM:MDMS$EXT_SCHEDULER.TEMPLATE
```

4.7 Removing SLS/MDMS V2.x Automatic Startup

If you have been using SLS/MDMS V2.x before and all your nodes running ABS and/or HSM now support the new MDMS make sure you remove this line from your system's start up file:

```
$ @SYS$STARTUP:SLS$STARTUP
```

If "SYS$MANAGER:TAPESYMBOL.COM" has been included in SYS$MANAGER:SYLOGIN.COM or the User Login Command Procedure, ensure that you remove this line from the same.

Note

You must have performed the conversion procedure as described before editing the TAPESTART.COM file.

4.8 Meeting OpenVMS Cluster Requirements

If you installed ABS server software on an OpenVMS Cluster system, perform the steps in Table 4–1 on each node in the OpenVMS Cluster (excluding the installing node) where you want to use ABS.

Note

The command line interface is not available with the ABS-OMT license, you must use the GUI.

Table 4–1 Updating the DCL Tables

Step	Action
1.	Run the command procedure ABS$STARTUP from each node that you want to use ABS. This ensures ABS$/MDMS$ logical names are defined the same across all nodes in the OpenVMS Cluster: `$ @SYS$STARTUP:ABS$STARTUP.COM`
2.	Update the DCL table on each node in the OpenVMS Cluster (excluding the installing node). Enter the following command on each node: `$ INSTALL REPLACE SYSLIBRARY:DCLTABLES.EXE`
3.	Have all system users log off and log on again to enable them to use the DCL ABS/MDMS commands (unless performing an upgrade).

4.9 Configure MDMS

Now that you have installed MDMS you need to configure the MDMS database. MDMS provides a command procedure that you can use to configure the MDMS database in a new installation. The procedure is completely self-documenting with a help facility, and all key object attributes are defined. The procedure guides you through the following configuration objects:

- The domain (defining default values for other objects)
- Locations
- Nodes and groups

- Jukeboxes and associated drives
- Standalone drives and stackers
- Volumes, media types and pools

The procedure is activated using the following command:

```
$ @ MDMS$SYSTEM:MDMS$CONFIGURE.COM
```

Please refer to the *Archive Backup System for OpenVMS Guide to Operations, Appendix* "Configuration Example" for an example of using this procedure. If you are upgrading from SLS or ABS V2.9x, you should use the procedure MDMS$CONVERT_V2_TO_V3 instead. Refer to Appendix D.

Once MDMS is installed, and any conversions are performed, you may wish to adjust your configuration prior to performing MDMS operations.

In addition to configuring MDMS, you may also have to configure robots for jukeboxes you are planning to use. For MRD-controlled jukeboxes, the robot name is the OpenVMS device name of the robot device, and they normally fall into one of several formats:

- GKx0 or GKxn01 for direct-connect SCSI
- nDUAnnn for access via an HSJ-type controller
- 2GGmx for Fibre Channel access

If the jukebox is controlled by direct-connect SCSI (first option), the robot device must first be loaded on the system with one of the following DCL commands:

```
Alpha – $ MCR SYSMAN IO CONNECT GKxxx/NOADAPTER/DRIVER=SYS$GKDRIVER.EXE

VAX – $ MCR SYSGEN CONNECT GKxxx/NOADAPTER/DRIVER=GKDRIVER
```

and the device name must begin with GK. It is suggested that these commands be included in a system startup file so that they are automatically configured at boot time.

4.10 Convert ABS Policy Database to MDMS Database

When upgrading from ABS/MDMS V3.n, you must convert the ABS Policy Database to MDMS Database. See Appendix D for information on how to do this procedure.

4.11 Configure Remote Tape Drives

If you installed the RDF software, you need to configure the remote tape drives. RDF is not available if you are installing MDMS with the ABS-OMT license.

For each tape drive served with RDF Server software, make sure there is a drive object record in the MDMS database that describes it. Refer to the chapters on MDMS configuration in the *Archive Backup System for OpenVMS Guide to Operations* and the MDMS CREATE DRIVE command in the command reference guide.

For each node connected to the tape drive, edit the file TTI_RDEV:CONFIG_node.DAT and make sure that all tape drives are represented in the file. See the *Archive Backup System for OpenVMS Guide to Operations* for more information on remote drive setup.

During startup of MDMS, the RDF client and server are also started. The RDF images are linked on your system. If you see the following link errors on Alpha V6.2, this is not an RDF bug. The problem is caused by installed VMS patches ALPCOMPAT_062 and ALPCLUSIO01_062.

```
%LINK-I-DATMISMCH, creation date of 11-FEB-1997 15:16 in
shareable image SYS$COMMON:[SYSLIB]DISMNTSHR.EXE;3
differs from date of 4-MAY-1995 22:33 in shareable image library
```

```
SYS$COMMON:[SYSLIB]IMAGELIB.OLB;1
.
.
.
```

This is a known problem and is documented in TIMA. To correct the problem, issue the following DCL commands:

```
$ LIBRARY/REPLACE/SHARE SYS$LIBRARY:IMAGELIB.OLB SYS$SHARE:DISMNTSHR.EXE
$ LIBRARY/REPLACE/SHARE SYS$LIBRARY:IMAGELIB.OLB SYS$SHARE:INIT$SHR.EXE
$ LIBRARY/REPLACE/SHARE SYS$LIBRARY:IMAGELIB.OLB SYS$SHARE:MOUNTSHR.EXE
```

4.12 Granting the Appropriate ABS/MDMS Access Right Identifiers

When ABS Server installation procedure is complete, the user account that performed the installation (typically the SYSTEM account) is granted the following ABS access rights identifiers. These identifiers are only needed if you use ABS DCL commands, or if you have the ABS_RIGHTS option turned on in the MDMS domain. For details on MDMS rights identifiers refer to Section 5.1 of ABS/MDMS Operations Guide.

- ABS_CREATE_STORAGE_CLASS–Users who are granted this access right identifier can create a storage class (applicable only on ABS server system).
- ABS_CREATE_EXECUTION_ENV–Users who are granted this access right identifier can create an execution environment (applicable only on ABS server system).
- ABS_SHOW_ALL–Users who are granted this access right identifier can show all ABS policy objects (applicable only on ABS server system).
- ABS_LOOKUP_ALL–Users who are granted this access right identifier can look up all ABS saved data from any catalog (applicable on any ABS node).
- ABS_CREATE_REMOTE_JOBS–Users who are granted this access right identifier can submit a save or restore request that will be executed on a remote client node (applicable only on ABS server system).

  Requirement:
  To create Windows or Tru64 UNIX save requests, the requester (creating process) must have the ABS_CREATE_REMOTE_JOBS access rights identifier enabled.
- ABS_BACKUP_JOB–Users who are granted this access right identifier can submit a save or restore request that runs under their user name (a user backup). The user must also have read and write access control to the storage policy and environment policy intended for user backups.
- ABS_BYPASS–Users who are granted this access right identifier can perform any ABS function (applicable only on ABS server system). This includes creating, deleting, modifying, or showing any ABS policy objects.

Note

Before any user can use MDMS commands or the MDMSView GUI, you must grant MDMS rights to those users. Enabling an Access Rights Identifier. See the *Archive Backup System for OpenVMS Guide to Operations* "Security" Chapter for more information on the use of identifiers.

To grant an access rights identifier to a user's account, run the AUTHORIZE utility.

Example:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE

UAF>GRANT/IDENTIFIER ABS_LOOKUP_ALL USER1

%UAF-I-GRANTMSG, identifier ABS_LOOKUP_All granted to USER1

UAF>EXIT
```

Note

Do not enable the access rights identifier by performing a SET RIGHTS_LIST at the DCL prompt.

4.12.1 Revoking An Access Rights Identifier

To remove an access right identifier, run AUTHORIZE utility and revoke the identifier from the user's account:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE

UAF>REVOKE/IDENTIFIER ABS_BYPASS USER1
%UAF-I-REVOKEMSG, identifier ABS_BYPASS revoked from USER1
```

You can modify the other policy objects provided by ABS in the same manner. See *Archive Backup System for OpenVMS Guide to Operations* for instructions about adding users and enabling access controls.

Before using your storage policy, you may need to modify the MDMS related information in the policy. For example, you may wish to use a different media type than the default media type from your MDMS domain. When ABS is installed, the storage policies are initialized with the defaults from the domain. Issue an MDMS SHOW DOMAIN command to see the defaults. Make sure that your storage policy contains the desired settings before executing a save request.

4.13 Verifying Windows and Tru64 UNIX Client Quotas

If you are supporting Windows or Tru64 UNIX clients, to ensure successful save and restore operations, set the quotas to the following values on ABS OpenVMS server node:

```
UCX> SET PROTOCOL TCP /QUOTA=(SEND:50000,RECEIVE:50000)
```

4.14 Adding Windows Parameter

Issuing multivolume SAVE requests to Windows client requires you to modify the Registry Path

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Sevices\Tcpip\Parameters
```

with the following Windows parameter (20 or greater is recommended).

```
TcpMaxDataRetransmissions     REG_DWORD      20
```

This change to the default built in Windows parameter/subkey ensures that the TCP/IP connection is not prematurely terminated with send failure.

Note

After making the changes to the parameter, you need to reboot the system to allow the changes to take effect.

4.15 Allowing ABS Access to All Files on the Windows Systems

ABS must have access to all the files you wish to backup on your Windows system. There are two ways to do this:

- Set access on the files for the SYSTEM account

  To set the file access,

  – Select the file from a fileview window.

  – Select Properties from the File pulldown.

  – Click the Security tab

  – Select Permissions.

  – Select Add and highlight SYSTEM. Add the type of access (full control is best, so you can restore files).

  – Click the Add button.

  This gives the SYSTEM account access to the files.

- Set the Account that ABS uses for the Backups

  ABS, by default, uses the SYSTEM account to backup the files. If you wish to change the account used by ABS, you may do this by modifying the properties of the service:

  – Click on Services in the Control Panel.

  – Highlight the ABS service

  – Click on Startup.

  – In the Log On As section, select This Account.

  – Enter the account name to be used and the password for that account.

  – Click OK.

  The account that you select should be a member of the Administrator group. The administrator group should be able to access all the files on your Windows system, unless you set access denied for Administrator on a file.

4.16 ABS/MDMS Graphical User Interface (GUI) Installation

This section describes how to install and run the Graphical User Interface (GUI) on various platforms.

As the GUI is based on Java, you must have the Java Runtime Environment (JRE) installed on the system where you are running the GUI. If you do not have the JRE installed on your system see the information below for kit locations.

The ABS/MDMS installation procedure extracts files from the kit and places them in MDMS$ROOT:[GUI...]. You can then move the Windows files to a Windows system and install them.

Note

For the GUI to communicate with the MDMS server, you must have TCP/IP services on the node where you have the MDMS server running.

The GUI requires the following in order to run:

Java Runtime Environment V1.2 or above - Since the ABS/MDMS GUI is a Java application, it requires the platform specific JRE. You can download the correct kit from the given URLs. For the OpenVMS installation, you may alternatively install the Standard Edition kit instead of the JRE kit. This is packaged as a PCSI kit, which is simpler to install. Issues concerning availability and installation can be directed to:

```
http://java.sun.com/products (for Microsoft Windows)

http://www.compaq.com/java (for OpenVMS Alpha)
```

Memory - On Windows systems, the hard drive space requirement is 4 MB for the MDMS GUI. The main memory space requirement for running the GUI is 10 MB.

4.16.1 Installing the GUI on OpenVMS Alpha

During the ABS/MDMS installation, the following question will be asked:

```
Do you want the MDMS GUI installed on OpenVMS Alpha [YES]?
```

Reply “Yes” to the question if you want to install the GUI on OpenVMS. Files will be moved to MDMS$ROOT:[GUI.VMS] and the GUI installation will be complete.

If you have *not* already installed the JRE, you should install it by following the instructions provided at the download site, http://www.compaq.com/java. The version specific setup command procedure provided by the Java installation will establish defaults for the logical names and symbols on the system. You should add a command line to the

```
SYS$COMMON:[SYSMGR]SYLOGIN.COM
```

command procedure to run this Java setup command procedure at login.

The JAVA$CLASSPATH is defined for the GUI in the

```
MDMS$SYSTEM:MDMS$START_GUI.COM
```

command procedure provided during the installation. The call to Java to invoke the GUI is also included in this command procedure.

Note

Make sure that the logical JAVA$USER_DCL is not defined or the GUI will not work.

4.16.2 Installing the GUI on Intel Windows NT/95/98

During the ABS/MDMS installation, the following question will be asked:

```
Do you want files extracted for Microsoft Windows [YES]?
```

Reply “Yes” if you want to install the GUI on a Microsoft Windows platform. Files will be moved to MDMS$ROOT:[GUI.WINTEL].

Move the file MDMS$ROOT:[GUI.WINTEL]SETUP.EXE to the Windows machine and run it to install the GUI.

If you have not installed the JRE, you should install it by following the instructions at the download site: http://java.sun.com/products.

4.17 Running the GUI

Now that you have installed the GUI, you have to make sure that the server node is configured to accept communications from the GUI. The server node for the GUI must have:

- TCP/IP enabled.
- The MDMS rights enabled on the SYSUAF record for the user.

To enable TCP/IP communications on the server, you have to set the TCP/IP Fullname attribute and enable the TCP/IP transport. See the command reference guide for information about setting these attributes on a node.

MDMS rights for the user must be enabled in the SYSUAF record so that the user may login to the server using the GUI. Refer to the command reference guide for information on MDMS rights.

The OpenVMS Alpha account which will be used to run the GUI should have the PGFLQUO account quota set to 200,000. Correspondingly, the system pagefile should contain enough space to handle the large pgflquo value. The account should also have a WSQUOTA value of at least 10,000. The sysgen parameter WSMAX should be 10,000 or higher. The system should also have a minumum of 128 meg of RAM, with a higher amount recommended.

4.17.1 Running the GUI on OpenVMS Alpha

To use the MDMS GUI on OpenVMS Alpha systems, use the following commands:

```
$ SET DISPLAY/CREATE/NODE=node_name/TRANSPORT=transport

$ MDMS/INTERFACE=GUI
```

For the SET DISPLAY command, the node_name is the name of the node on which the monitor screen exists. This allows you to use the GUI on systems other than those running OpenVMS Alpha V7.2 or higher. The transport must be a keyword of:

- LOCAL - if you are running the GUI on the same node as the monitor.
- DECNET - if you are running the GUI on a monitor connected to another node and you wish to use DECnet protocol between the monitor node and the GUI Java node.
- TCPIP - if you are running the GUI on a monitor connected to another node and you wish to use TCP/IP protocol between the monitor node and the GUI Java node.

4.17.1.1 Running the GUI on Windows NT/95/98

To use the ABS/MDMS GUI on Microsoft Windows platforms, double click on the batch file named MDMSView.bat in the MDMSView directory on the drive where you installed the GUI.

To view the ABS/MDMS GUI correctly, you must have the display property settings for the screen area set to 1024 X 768 or higher.

If you have Java installed in a location other than the normal default location, you will need to first edit the MDMSView.bat file to include the correct path. The default in this file is

```
C:\Program Files\JavaSoft\JRE\1.2\bin\java.exe
```

4.18 How to Remove ABS/MDMS to Version 4.0 Files

Rolling back ABS/MDMS to V3.2A is only to be used in emergency situations. Any save requests done since the upgrade will be lost. Any MDMS database changes done since the upgrade will be lost. If tape volumes have been allocated, deallocated, or reused since the upgrade, these will not be reflected in the volume database after the rollback. Please use caution when deciding to do a rollback.

Note

Prior to upgrading to ABS/MDMS V4.0, to be able to get back to the exact V3.2A situation, you were at before the upgrade, your ABS catalogs would have to be backed up while in a quiet state. Then, you could restore the V3.2A catalogs after the rollback, and they would not reflect any saves run after the V4.0 upgrade. If you do not do this, then there will be entries in the catalog which reference tapes which may be in a free state in the old MDMS database files.

If you need to remove V4.0 files prior to reinstalling V3.2A, follow this procedure:

- Shutdown ABS and MDMS by executing SYS$STARTUP:ABS$SHUTDOWN.COM and SYS$STARTUP:MDMS$SHUTDOWN.COM.
- Execute the following command procedure:

  ```
  $ @ABS$SYSTEM:ABS$REMOVE_V40_FILES.COM
  ```

- Reinstall ABS/MDMS V3.n. Be sure that you do NOT use the STANDARD installation method and that you install both ABS and MDMS. You may optionally restore backups of your ABS catalogs (see note above).

After the reinstall of ABS/MDMS V3.n, ABS and MDMS should be running as they were before the V4.0 installation. If any operations have been done (ie. moving a volume, creating a new save request), these will not be reflected in the restored database files.

4.19 Configuration of ABS/MDMS for System Backup to Tape for Oracle Databases

If you are configuring ABS/MDMS for System Backup to Tape for Oracle Databases, refer to the System Backup to Tape for Oracle Databases section in the *Archive Backup System for OpenVMS Guide to Operations.*

A

Examples of Authorizing Windows and Tru64 UNIX Clients

This appendix contains examples of how to authorize Windows and Tru64 UNIX clients. This includes adding, modifying, showing, and removing Windows and Tru64 UNIX client licenses.

To use the license command as shown in the example in this appendix, you can define the following symbol at the system prompt:

```
$ LICENSE := $SYS$SYSTEM:ABS$CLIENT_LICENSE.EXE
```

All of the examples in the following sections use this symbol definition.

A.1 Adding Client Licenses

Example A–1 shows how to add a Tru64 UNIX or Windows client license

Example A–1 Adding Client Licenses

```
$ LICENSE

Would you like to Add/Modify/Remove/Show the Client License?: ADD
Enter Node Name: NTNODE
Client Node Type (UNIX or NT) [UNIX]: NT
Enter TCPIP Port Number [1800]: 1800
Client NTNODE successfully ADDED to License Database
License Count: 1 used of 6 total

$ LICENSE

Would you like to Add/Modify/Remove/Show the Client License?: ADD
Enter Node Name: UNIX_1
Client Node Type (UNIX or NT) [UNIX]: UNIX
Client UNIX_1 successfully ADDED to License Database
License Count: 1 used of 25 total
```

A.2 Modifying Client Licenses

Example A–2 shows how to modify the port number of an Windows or Tru64 UNIX client license.

Note

The port number that you enter here must match the port number you assigned during the NT client installation.

Example A–2 Modifying a Client License

```
$ LICENSE

Would you like to Add/Modify/Remove/Show the Client License?: MODIFY
Enter Node Name: NTNODE
Client Node Type (UNIX or NT) [UNIX]: NT
Current values
Type: Windows NT
Transport:TCP/IP
Port:1800

New Port #?: 1800
Client NTNODE successfully MODIFIED in License Database

License Count: 1 used of 6 total

$ LICENSE

Would you like to Add/Modify/Remove/Show the Client License?: MODIFY
Enter Node Name: UNIX_1
Client Node Type (UNIX or NT) [UNIX]: UNIX
Current values
Type: UNIX
Transport: TCP/IP
Port: 514

New Port #?: 1800
Client UNIX_1 successfully MODIFIED in License Database

License Count: 1 used of 25 total
```

A.3 Showing Client Licenses

Example A–3 illustrates how to show an Windows or Tru64 UNIX client license.

Example A–3 Showing Client Licenses

```
$ LICENSE

Enter Node Name or [ALL]: ALL
Client Node Type (UNIX or NT) [UNIX]:  NT

Node Name           Type         Transport    Port
---------           ----         ---------    ----
NTNODE              Windows NT    TCP/IP      1800

License Count: 1 used of 6 total

$ LICENSE

Enter Node Name or [ALL]: ALL
Client Node Type (UNIX or NT) [UNIX]:  UNIX

Node Name           Type         Transport    Port
---------           ----         ---------    ----
UNIX_1              UNIX          TCP/IP      1800

License Count: 1 used of 25 total
```

A.4 Removing Client Licenses

Example A–4 shows how to remove Windows or Tru64 UNIX client licenses:

Example A–4 Removing Client Licenses

```
$ LICENSE

Would you like to Add/Modify/Remove/Show the Client License?: REMOVE
Enter Node Name: NTNODE
Client Node Type (UNIX or NT) [UNIX]: NT
Client NTNODE successfully REMOVED from License Database
License Count: 0 used of 6 total
```

B

Upgrading ABS From the ABS-OMT License

When upgrading ABS after using the ABS-OMT license, you only have to perform a couple of steps. Follow these steps, to upgrade ABS:

1. Shutdown ABS by issuing the command

   ```
   $ @SYS$MANAGER:ABS$SHUTDOWN.COM
   ```

2. Register and load the ABS-OMT-UPG license. You need both the ABS-OMT and ABS-OMT-UPG license.

3. Startup ABS by issuing the command:

   ```
   $ @SYS$STARTUP:ABS$STARTUP.COM
   ```

4. Run the following executable to load the default ABS storage and environment policies:

   ```
   $ RUN SYS$SYSTEM:ABS$DB_INIT.EXE
   ```

   All features of ABS are now enabled.

C

How to Delete ABS/MDMS From Your System

To delete ABS software from your system, shut down ABS and delete it from the system:

```
$ @SYS$STARTUP:ABS$SHUTDOWN
$ @ABS$SYSTEM:ABS$DELETE_ABS
```

To delete MDMS software from your system, shut down MDMS and uninstall it from your system:

```
$ @SYS$STARTUP:MDMS$SHUTDOWN
$ @SYS$STARTUP:MDMS$UNINSTALL.COM
```

D

Converting to ABS/MDMS V4.0

D.1 Introduction

This appendix discusses the various conversion activities required when updating to ABS/MDMS V4.0.

SLS V2.x --> ABS/MDMS V4.0

SLS V2.x uses tapestart.com, volume and magazine databases, various data files, and SBK files to do backups. To go to ABS/MDMS V4.0, you may wish to convert your media information to MDMS databases and your SBK files into ABS/MDMS objects. Or you may decide to create all new information in ABS/MDMS V4.0 and skip the conversions. You may do one, both, or none of the conversions.

- If you wish to convert your tapestart.com, volume and magazine databases into MDMS V4.0, use the command procedure MDMS$SYSTEM:MDMS$CONVERT_V2_TO_V3. See Appendix D.2
- If you wish to convert SLS SBK files into ABS/MDMS V4.0 objects, you may use the command procedure ABS$SYSTEM:SLS_CONVERT.COM. See Appendix D.3.

ABS/MDMS V2.x --> ABS/MDMS V4.0

ABS V2.x uses tapestart.com, volume and magazine databases, and various data files for media management and an ABS policy database for the ABS objects. You may wish to convert the media information into the MDMS databases or you may create new objects. In ABS/MDMS V4.0 the ABS Policy Engine has been moved into the MDMS Server. To upgrade to ABS V4.0 the ABS 2.x or 3.x Policy Database information must be exported to the MDMS Database. There are also some catalog modifications which must be done.

Note

If you are using an ABS Rdb policy database, this must be converted to an RMS database before exporting the data to V4.0 format. The Rdb conversion must be done before updating to V4.0. See Appendix D.5

- If you wish to convert your tapestart.com, volume and magazine databases into MDMS V4.0, use the command procedure MDMS$SYSTEM:MDMS$CONVERT_V2_TO_V3. See Appendix D.2.
- Convert your ABS catalogs using SYS$SYSTEM:ABS$CATALOG_UPGRADE.EXE. See Appendix D.4.
- Convert the ABS Rdb database to an RMS database. See Appendix D.5.
- Convert the ABS policy database to V4.0 format using the image SYS$SYSTEM:ABS$CONVERT_V3_TO_V4.EXE. See Appendix D.6.

ABS V3.0B and MDMS 2.x--> ABS/MDMS V4.0

MDMS V2.x uses tapestart.com, volume and magazine databases. You may wish to convert them to the MDMS databases or create new objects. ABS V3.0B uses the ABS policy databases which must be moved to the MDMS database.

– If you wish to convert your tapestart.com, volume and magazine databases into MDMS V4.0, use the command procedure MDMS$SYS-TEM:MDMS$CONVERT_V2_TO_V3. See Appendix D.2.

– Convert the ABS policy database to V4.0 format using the image ABS$SYS-TEM:ABS$CONVERT_V3_TO_V4.EXE. See Appendix D.6.

ABS/MDMS V3.1x or 3.2x--> ABS/MDMS V4.0

MDMS V3.x needs no conversions to work with ABS/MDMS V4.0. ABS V3.1 or V3.2 use the ABS policy database which must be moved into the MDMS database.

– Convert the ABS policy database to V4.0 format using the image ABS$SYS-TEM:ABS$CONVERT_V3_TO_V4.EXE. See Appendix D.6.

D.2 Converting SLS/MDMS V2.X Symbols and Database

This section describes how to convert the SLS/MDMS V2.X symbols and database to Media and Device Management Services Version 4 (MDMS). The conversion is automated as much as possible, however, you will need to make some corrections or add attributes to the objects that were not present in SLS/MDMS V2.X.

Before doing the conversion, you should read the chapter about Media Management in the *Archive Backup System for OpenVMS Guide to Operations* to become familiar with configuration requirements.

All phases of the conversion process should be done on the first database node on which you installed MDMS V4. During this process you will go through all phases of the conversion:

1. Convert the symbols in SYS$STARTUP:TAPESTART.COM into the following objects:

- Locations
- Domain
- Nodes
- Media types
- Jukeboxes
- Drives

2. Convert the database authorization file, VALIDATE.DAT, into node objects.

- Convert the rest of the database files:
- Pool Authorization file (POOLAUTH.DAT)
- Slot Definition file (SLOTMAST.DAT)
- Volume Database file (TAPEMAST.DAT)
- Magazine Database file (SLS$MAGAZINE_MASTER_FILE.DAT)

When you install on any other node that does not use the same TAPESTART.COM as the database node, you only do the conversion of TAPESTART.COM

D.2.1 Executing the Conversion Command Procedure

To execute the conversion command procedure, type in the following command:

```
$ @MDMS$SYSTEM:MDMS$CONVERT_V2_TO_V3
```

The command procedure will introduce itself and then ask what parts of the SLS/MDMS V2.x you would like to convert.

During the conversion, the conversion program will allow you to start and stop the MDMS server. The MDMS server needs to be running when converting TAPESTART.COM and the database authorization file. The MDMS server should not be running during the conversion of the other database files.

During the conversion of TAPESTART.COM the conversion program generates the following file:

```
$ MDMS$SYSTEM:MDMS$LOAD_DB_nodename.COM
```

This file contains the MDMS commands to create the objects in the database. You have the choice to execute this command procedure or not during the conversion.

The conversion of the database files are done by reading the SLS/MDMS V2.x database file and creating objects in the MDMS V4 database files.

You must have the SLS/MDMS V2.x DB server shutdown during the conversion process. Use the following command to shut down the SLS/MDMS V2.x DB server:

```
$ @SLS$SYSTEM:SLS$SHUTDOWN
```

D.2.2 Resolving Conflicts During the Conversion

Because of the difference between SLS/MDMS V2.x and MDMS V4 there will be conflicts during the conversion. Instead of stopping the conversion program and asking you about each conflict, the conversion program generates the following file during each conversion:

$ MDMS$MDMS$LOAD_DB_CONFLICTS_nodename.COM

Where nodename is the name of the node you ran the conversion on. This file is not meant to be executed, it is there for you to look at and see what commands executed and caused a change in the database. This change is flagged because there was already an object in the database or this command changed an attribute of the object.

An example could be that you had two media types of the same name but one specified compressed and one other specified non compressed. This would cause a conflict. MDMS V4 does not allow two media types with the same name but different attributes. What you see in the conflict file would be the command that tried to create the same media type. You will have to create a new media type.

Table D–1 shows the symbols in TAPESTART.COM file and what conflicts they may cause.

At the completion of the conversion of the database files, you will see a message that notes the objects that where in an object but not defined in the database. For example: the conversion program found a pool in a volume record that was not a pool object.

Table D–1 Symbols in TAPESTART.COM

TAPESTART.COM Symbol	MDMS V4 Attribute or Object	Possible conflict
ALLOCSCRATCH	If defined, adds the SCRATCH_TIME attribute to the domain object.	If the ALLOCSCRATCH symbol is different in different TAPESTART.COM files a line is added to the conflict file.
DB_NODES	If defined, creates a node object for the nodes in the DB_NODES list.	A conflict can be generated if the node exists and an attribute changed with a different TAPESTART.COM file. Every drive and jukebox definition in the TAPESTART.COM can cause a node to be created with a /NODATABASE_SERVER qualifier. A DB node will change the attribute to database server, this can cause a line to be added to the conflict file.
DCSC_n_NODES	If defined, creates a node object and adds the node attribute to the DCSC jukebox.	All adds of nodes to jukeboxes cause a line to be added to the conflict file.
DCSC_DRIVES	If defined, creates a drive object for DCSC.	If an attribute is different when adding attributes a line is added to the conflict file.
DENS_x	If defined, adds the density or compaction attribute to a media type. If the value is COMP or NOCOMP then the compaction attribute is define: YES or NO. If the density is anything other than COMP or NOCOMP then the value is placed in the density attribute.	A line is added to the conflict file if the DENS_x is different.
FRESTA	If defined, adds the deallocate state attribute to the domain object.	If the FRESTA symbol is different in different TAPESTART.COM files a line is added to the conflict file.
LOC	Creates a location object and also sets the ONSITE_LOCATION attribute in domain object.	If the object exists or is different than the onsite location attribute in the domain object a line to be added to the conflict file. This can happen when you have different LOC symbol in two TAPESTART.COM files.
MAXSCRATCH	If defined, adds the maximum scratch time attribute to the domain object.	If the MAXSCRATCH symbol is different in different TAPESTART.COM files a line is added to the conflict file.

Table D–1 Symbols in TAPESTART.COM

TAPESTART.COM Symbol	MDMS V4 Attribute or Object	Possible conflict
MTYPE_x	Creates a media type object for each MTYPE_x.	A line is added to the conflict file if a media type is already in the database and another one has the same name. In SLS/MDMS V2.x you could have the same media type name with compaction and nocompaction. In MDMS you cannot have two media types with the same name. You need to change the name of one of the media type and enter it into the database. You will also have to change ABS or HSM to reflect this. Also, you may have to change volume and drive objects.
NET_REQUEST_TIMEOUT	If defined, adds the NETWORK_TIMEOUT attribute to the domain object.	If the NET_REQUEST_TIMEOUT is different in different TAPESTART.COM files a line is added to the conflict file.
PROTECTION	Adds the default protection to the domain object.	A line is added to the conflict file if the protection is changed.
QUICKLOAD	When drives are created, this attribute is added as automatic reply.	A line is added to the conflict file if a drive's automatic reply is changed.
TAPE_JUKEBOXES	Creates a jukebox object for each jukebox in the list.	A line is added to the conflict file if a jukebox is already defined and any of the attributes change.
TAPEPURGE_MAIL	If defined, adds the mail attribute to the domain object.	If the TAPEPURGE_MAIL is different in different TAPESTART.COM files a line is added to the conflict file.
TOPERS	If defined, adds the Opcom class attribute to the domain object.	If the TOPERS symbol is different in different TAPESTART.COM files a line is added to the conflict file.
TRANS_AGE	If defined, adds the transition time attribute to the domain object.	If the TRANS_AGE symbol is different in different TAPESTART.COM files a line is added to the conflict file.
VLT	Creates a location object and also sets the OFFSITE_LOCATION attribute in domain object.	If the object exists or is different than the offsite location attribute in the domain object, a line is added to the conflict file. This can happen when you have different VLT symbol in two TAPESTART.COM files.

D.2.3 Things to Look for After the Conversion

Because of the differences between SLS/MDMS V2.x and MDMS V4 you should go through the objects and check the attributes and make sure that the objects have the attributes that you want.

Table D–2 shows the attributes of objects that you may want to check after the conversion.

Table D–2 Things to Look for After the Conversion

Object	Attribute	Description
Drive	Drive	Make sure you have all of the drives defined. In the MDMS V4 domain, you can only have one drive with a given name. In SLS/MDMS V2.x you could have two drives with the same name if they were in different TAPESTART.COM files. You should make sure that all drives in your domain are in the database. You may have to create one drive with a name of say, DRIVE1 with a device name of 1MUA520 and a node of NODE1. Then create another drive, DRIVE2, with a device name of 1MUA520 and a node of NODE2. A line is added to the conflict file every time a node is added to a drive. This flags you to check that the node really belongs to this drive of if you need to create another drive.
	Description	Make sure this is the description you want for this drive. This attribute is not filled in during the conversion program.
	Device	Make sure this device name does not have a node name as part of it.
	Nodes	Make sure this list of nodes contains the nodes that can reach this drive.
	Disabled	The conversion program enables all drives. If you want this drive disabled, then set this attribute to YES.
	Shared	The conversion program sets this attribute to NO. NO means that MDMS does not have to compete with other applications for this device. If MDMS is supposed to share this device with other applications set this attribute to YES.
	State	Make sure this drive is in the right state. If the drive is not in the right state, you can set this attribute to the right state or issue the following command: $ MDMS SET DRIVE drive/CHECK
	Automatic reply	The conversion program sets this attribute from the QUICK-LOAD symbol. Make sure this is the way you want the drive to react.
	RW media types	The conversion program added media types to this drive as it found them. Make sure these are the correct read-write media types for this drive.
	RO Media Types	There are no read-only media types in SLS/MDMS V2.x so none is added to the drives during conversion. You may want to add some read-only media types to the drive object.
	Access	The conversion program has no way of knowing what the access should be, therefore, it sets the access attribute to ALL. Make sure this is the access you want for this drive.
	Jukebox	Make sure this is the jukebox that this drive is in.
	Drive Number	Make sure this is the drive number for robot commands.

Table D–2 Things to Look for After the Conversion

Object	Attribute	Description
Domain	Description	Make sure this is the description you want for your domain. The default is: Default MDMS Domain.
	Mail	Make sure this is where you want mail sent when a volume reaches its scratch data and MDMS dellocates it. If you do not want mail sent, make the value blank. The default is: SYSTEM.
	Offsite location	Make sure this is the offsite location that you want for the default when you create objects. This was set to the value of VLT from TAPESTART.COM file. This could be different in each TAPESTART.COM file.
	Onsite location	Make sure this is the onsite location that you want for the default when you create objects.
	Default media type	Make sure this is the media type you want assigned to volumes that you do not specify a media type for, while creating.
	Deallocate state	Make sure this is the default state you want volumes to go to after they have reached their scratch date. This could be changed each time that you convert the TAPESTART.COM file on a new node.
	Opcom classes	Make sure these are the Opcom classes where you want MDMS OPCOM messages directed. This could be changed each time you convert the TAPESTART.COM file on a new node.
	Protection	Make sure this is the default protection that you want assigned to volumes that you do not specify a protection for.
	Maximum scratch time	Make sure this is the default maximum scratch time you want for volumes in your domain. This could be changed each time that you convert the TAPESTART.COM file on a new node.
	Scratch time	Make sure this is the default scratch time you want for volumes in your domain. This could be changed each time that you convert the TAPESTART.COM file on a new node.
	Transition time	Make sure this is the default transition time you want for volumes in your domain. This could be changed each time that you convert the TAPESTART.COM file on a new node.
	Network time-out	Make sure this is the network timeout you want. This could be changed each time that you convert the TAPESTART.COM file on a new node.
Location	Description	Make sure this is the description you want for this location. This attribute is not filled in during the conversion program.
	Spaces	The conversion program cannot fill in spaces so make sure you set the spaces attribute.
	In location	The conversion program cannot fill in this attribute so make sure if this location is in a higher level location you set this attribute.

Table D–2 Things to Look for After the Conversion

Object	Attribute	Description
Media type	Media type	Make sure you have all the media types that you had before. In the MDMS V4 you can only have on media type with the same name. If you had two media types in SLS/MDMS V2.x with the same name, the second one is not created in the MDMS V4 database.
	Description	The conversion program does not add a description to this attribute. Type in a description for this attribute.
	Density	The density attribute is only changed when the DENS_x symbol in the TAPESTART.COM file is something other than COMP or NOCOMP. Check to make sure this is correct.
	Compaction	This attribute is set to YES if the DENS_x symbol in the TAPESTART.COM file is COMP. It is set to NO if the symbol is NOCOMP. Check to make sure this is right.
	Capacity	This attribute is set to the value of DENS_X from the TAPESTART.COM file if it is not defined as COMP or NOCOMP. Check to make sure this right.
Jukebox	Description	The conversion program does not put a description for this attribute. Type in a description to this attribute.
	Nodes	Make sure this list of nodes contains the nodes that can reach the robot.
	Location	Make sure this is the location where this jukebox is located.
	Disabled	The conversion program enables all jukeboxes. If you want this jukebox disabled, set this attribute to YES.
	Shared	The conversion program sets this attribute to NO. NO means that MDMS does not expect to compete with other applications for this jukebox. If MDMS is supposed to share this jukebox with other applications set this attribute to YES.
	Auto reply	The conversion program sets this attribute to NO. Make sure this is the way you want the jukebox to react.
	Access	The conversion program has no way of knowing what the access should be, therefore, it sets the access attribute to ALL. Make sure this is the access you want for this jukebox.
	Control	Make sure that the attribute is set to MRD if MRD controls the robot. If the robot is controlled by DCSC, this attribute should be set to DCSC.
	Robot	Make sure this is the robot for this jukebox.
	Slot count	You need to set the slot count. The conversion program has no way of finding out the slot count.
	Usage	Make sure the usage is correct for the type of jukebox you have. The conversion program has no way of finding out if the jukebox uses magazines or not. If this jukebox uses magazines, you will need to configure it.
Magazine	Description	The conversion program does not add a description to this attribute. Type a description for this attribute.

Table D–2 Things to Look for After the Conversion

Object	Attribute	Description
	Offsite location	The old magazine record had no offsite location, so you need to add this attribute.
	Offsite date	The old magazine record had no offsite date, so you need to add this attribute.
	Onsite location	The old magazine record had no onsite location, so you need to add this attribute.
	Offsite date	The old magazine record had no onsite date, so you need to add this attribute.
Node	Description	The conversion program does not put a description in this attribute. Type a description for this attribute.
	DECnet-Plus fullname	TAPESTART.COM does not support DECnet-Plus, therefore the conversion program cannot put in the DECnet-Plus fullname attribute. If this node uses DECnet-Plus, you should set this attribute.
	TCP/IP fullname	TAPESTART.COM does not support TCP/IP, therefore the conversion program cannot put in the TCP/IP fullname attribute. If this node uses TCP/IP, you should set this attribute.
	Disabled	The conversion program sets the enabled attribute. Make sure you want this node enabled.
	Database server	If this attribute is set to YES, this node has the potential to become a database server. The logical MDMS$DATABASE_SERVERS must have this node name in is definition of nodes in the domain. This definition is defined in the SYS$STARTUP:MDMS$SYSTARTUP.COM file
	Location	Make sure this is the location that this node is located in. During the conversion it could have been changed depending on the TAPESTART.COM file or what the default was in the domain object at the time of creation.
	Opcom classes	This attribute is defined as the Opcom class in the domain object when the node was created. Make sure this is the Opcom class for this node.
	Transports	Make sure this is the transport you want. The conversion program has no way of knowing the transports you want so it takes the defaults.
POOL	Description	Make sure this is the description you want for this pool. This attribute is not filled in during the conversion program.
	Authorized users	Make sure that the comma separated list contains all of the authorized users for the pool. The users must be specified as NODE::user
	Default users	You need to set this attribute because conversion program does not set this attribute. The users must be specified as node::user.
VOLUME		The conversion program fills in all needed attributes from the old record. This is included so you will not think the volume object was forgotten.

D.2.4 Upgrading the Domain to MDMS V4

Upgrading your SLS/MDMS V2 domain starts with the nodes, which have been defined as database servers in symbol DB_NODES in file TAPESTART.COM. Refer to the Installation Guide for details on how to perform the following steps.

Step 1. Shut down all SLS/MDMS database servers in your SLS/MDMS domain.

Step 2. Install version MDMS V4 on nodes, which have been acting as database servers before.

Step 3. When the new servers are up-and-running check and possibly change the configuration and database entries so that it matches your previous SLS/MDMS V2 setup

Step 4. Edit SYS$MANAGER:MDMS$SYSTARTUP.COM and make sure that:

- Logical name MDMS$DATABASE_SERVERS include this nodes DECnet (Phase IV) node name
- Logical name MDMS$PREV3_SUPPORT is set to TRUE to enable the SLS/MDMS V2 support function in the new server
- Logical name MDMS$VERSION3 is set to TRUE to direct ABS and/or HSM to use the new MDMS V4 interface

If you had to change any of the logical name settings above you have to restart the server using '@SYS$STARTUP:MDMS$STARTUP RESTART'. You can type the server's logfile to verify that the DECnet listener for object SLS$DB has been successfully started.

Step 5. To support load, unload and operator requests from old SLS/MDMS clients you have to edit SYS$MANAGER:TAPESTART.COM and change the line which defines DB_NODES to read like this:

```
$ DB_NODES = ""
```

This prevents a SLS/MDMS V2 server from starting the old database server process SLS$TAPMGRDB.

Step 6. Now you are ready to start up ABS or HSM.

D.2.5 Convert from MDMS Version 3 to a V2.X Volume Database

This section describes how to convert the MDMS V4 volume database back to a SLS/MDMS V2.X volume database.

If for some reason, you need to convert back to SLS/MDMS V2.X a conversion command procedure is provided. This conversion procedure does not convert anything other than the volume database. If you have added new objects, you will have to add these to TAPESTART.COM or to the following SLS/MDMS V2.X database files:

- database authorization file (VALIDATE.DAT)
- Pool Authorization file (POOLAUTH.DAT)
- Slot Definition file (SLOTMAST.DAT)
- Volume Database file (TAPEMAST.DAT)
- Magazine Database file (SLS$MAGAZINE_MASTER_FILE.DAT)

To execute the conversion command procedure, type in the following command:

```
$ @MDMS$SYSTEM:MDMS$CONVERT_V3_TO_V2
```

After introductory information, this command procedure will ask you questions to complete the conversion.

D.3 SLS V2.x to ABS V4.0 Conversion Process

This chapter identifies the Conversion Process from SLS to ABS. First, the steps involved in converting from SLS to ABS are presented and explained. A Conversion Utility to help with the conversion is then presented.

D.3.1 Steps for Conversion

This section identifies the steps involved in converting a site's backup management from SLS to ABS. These steps are intended as guidelines, since each site has different requirements and need for their backup management.

D.3.1.1 Convert the MDMS Database

The first step in converting from SLS to ABS is to convert the volume, slot and magazine databases, and the media and device portions of TAPESTART.COM to MDMS databases. A command procedure is provided for this purpose and this procedure is documented in Appendix D.2 . Please note that this version of ABS and all future versions require the accompanying version of MDMS included in the installation kit.

D.3.1.2 Determine your use of SLS

The next step in converting from SLS to ABS is to identify how you use SLS. There are three major uses of SLS:

- System Backups
- Standby Archiving
- User Backups

ABS provides the same functionality as SLS System Backups and SLS User Backups. However, ABS cannot perform the same function as SLS Standby Archiving.

If you use SLS System Backups (as many sites do), then converting to ABS is fairly simple. If you use SLS User Backups, converting to ABS is slightly more involved, but is still fairly straightforward. If you use SLS Standby Archiving, ABS will not provide equivalent functionality.

D.3.1.3 Converting SLS System Backups to ABS

This section describes how to convert SLS System Backups (SBK files) into ABS Policy.

Determine valid SBK files

At many sites, only a few of the SBK files which reside in SLS$SYSBAK are actually used. The other SBK files are a result of experimentation, false starts at configuring SLS, or are obsolete.

In order to simplify the conversion of SLS System Backups, you should first identify the SBK files which you actually use. Usually, SBK Files are used at your site if:

- They are scheduled regularly by SLS
- They are manually executed by you or the Operator

A simple way of finding the SBK files which are scheduled by SLS is to search the SBK files for the DAYS_1 symbol. Any SBK file which does not define DAYS_1, or defines it as blank, is not scheduled by SLS for automatic execution. These SBK files are prime candidates for obsolete or unused files.

After identifying the SBK files which are not automatically scheduled, carefully determine which of the files may be invoked manually by you or the Operator.

Once you have identified the obsolete or unused SBK files, you can remove them from SLS$SYSBAK (after backing them up in case of mistakes, of course).

Convert the Valid SBK Files to ABS Policy

Once you have cleaned up your SLS$SYSBAK directory to only contain those SBK files you actually use, it is time to convert these SBK files to ABS Storage Classes (Archives), Execution Environments and Save Requests.

Compaq provides a utility to help in this conversion process. The conversion utility is called SLS_CONVERT.COM, and is included as an installation kit on the ABS Kit. To install the SLS to ABS Conversion utility, issue the command:

```
$ @SYS$UPDATE:VMSINSTAL   SLSTOVABS041 ABS$SYSTEM:! VAX System OR
$ @SYS$UPDATE:VMSINSTAL SLSTOAABS041 ABS$SYSTEM:! Alpha System
```

This command will install the conversion utility into ABS$SYSTEM:SLS_CONVERT.COM, and will create a subdirectory under the ABS$ROOT called SLS_CONVERSION. In addition, a logical name, ABS$SLS_CONVERSION will be defined to point to the work directory for the conversion effort.

The conversion utility creates DCL Command Procedures which issues appropriate ABS DCL commands equivalent to each SBK file. No changes are made to your ABS Policy Configuration directly. This allows you to experiment with the conversion utility safely, without affecting either the execution of your SLS SBK files, or starting ABS Save Requests inadvertently.

Once you have installed the conversion utility, you can create ABS command procedures for all of your SBK files by issuing the command:

```
$ @ABS$SYSTEM:SLS_CONVERT   *
```

The asterisk indicates you want to convert all SBK files to ABS DCL Commands. If you only want to convert one SBK file, you can specify the name of the SBK without the _SBK.COM or SLS$SYSBAK on the command line. For example, to convert NIGHTLY_SBK.COM, you would issue the command:

```
$ @ABS$SYSTEM:SLS_CONVERT NIGHTLY
```

Evaluate the ABS Conversion Command Files

After running the conversion utility, the ABS$SLS_CONVERSION directory will contain one ABS DCL Command Procedure for each SBK file converted. These output command files will contain:

- Comments explaining the conversion process
- ABS DCL Commands to create ABS Policy Objects
- A Prolog and Epilog command to complete the functions performed by the SBK file

You should not execute these command procedures blindly. The conversion utility attempts to duplicate the backup policy reflected in each SBK file, but you should carefully examine each command file produced to ensure that errors were not made, and that the ABS Policy to be created correctly reflects the backup policy you expect.

The things you should check for in the produced command procedures are:

- Naming conventions used in the conversion may not be what you want
- Errors in converting the SBK policy
- Possible ABS Policy Consolidation

The command procedure for each SBK file processed will contain the ABS DCL Commands to create one Storage Class (Archive), one Execution Environment and one or more Save Requests.

Naming Conventions Used

The name of the Storage Class (Archive) created for an SBK file will be the value of the CONTINUE symbol (if defined) followed by the suffix "_SC". If the CONTINUE symbol is not defined, the Storage Class (Archive) will be named the same as the SBK file with the "_SC" suffix.

The Environment created for an SBK will be named the same as the SBK file, but with an "_ENV" suffix. When a Save Request specifies a Storage Class (Archive) , the default Environment used will be the same name as the Storage Class (Archive), but with the "_ENV" suffix. Thus, the Environment created should be used by default.

Each SBK File will produce one or more ABS Save Requests. More than one ABS Save Request will be produced from an SBK file if all the following conditions are met:

- There are more than one FILES_n is specified
- The QUALIFIERS_n differ in the type of operation
- More than eight include specifications are found in a single SBK file (This limit has been removed in ABS/MDMS V4.0, so you may want to include more than 8 includes in a save).

For example, if an SBK file has QUALIFIER_1 defined as "/IM" indicating an Image (or Full) backup operation, but QUALIFIERS_2 is defined as "/SINCE=BACKUP" indicating an Incremental operation, then ABS will need two separate Save Requests to implement this policy. This is because an ABS Save Request will only do Full, Incremental or Selective operations, not a mix of them.

If all QUALIFIERS_n specify the same movement type and there are fewer than eight FILES_n, then ABS can combine all the operations into a single Save Request.

The Save Requests created will be named the same as the SBK file, but with "_FULL", "_INC" or "_SEL" to indicate the data movement type included in the Save Request. For example, if the SBK file NIGHTLY_SBK.COM defines FILES_1 through FILES_20, and all qualifiers include the "/IM" Image qualifier, then the conversion tool will create three Save Requests, called NIGHTLY_FULL_1 through NIGHTLY_FULL_3. Because ABS has a limit of 8 operations per save request, NIGHTLY_FULL_1 and NIGHTLY_FULL_2 would perform 8 Full backup operations, and NIGHTLY_FULL_3 would perform the last four. This limit has been removed in ABS/MDMS V4.0, so you may wish to include more operations per save request.

Consolidate ABS Policy Objects

The Conversion Utility shipped with ABS is a very simple utility. It converts each SBK file into the appropriate ABS DCL Commands to create the Storage Classes (Archives), Execution Environments and Save Requests necessary to reflect the backup policy in the SBK file.

No attempt is made to consolidate the Storage Classes (Archives) and Execution Environments, or to overlay the Save Requests for more optimum performance.

Before executing the command procedures to create the ABS Policy objects, you should try to consolidate Storage Classes (Archives) and Execution Environments. Save Requests may be combined if warranted by the intended policy, but in some cases, breaking a Save Request into several is better for reducing nightly backup time, simplifying an overall backup policy, or backing up different objects at different intervals.

Consolidating Storage Classes (Archives)

Consolidating the Storage Classes (Archives) is done by comparing their parameters. For each pair of Storage Classes (Archives), you can determine whether they can be combined by using the Table D–3. Note that in all cases, you can decide that one or the other parameter is correct for *both*, and consolidate based upon that decision.

Table D–3 Storage Class (Archive) Parameter

Storage Class (Archive) Parameter	Matching Criteria
Name	Doesn't matter choose meaningful name
Media Type	Should match
Tape Pool	Should match
Media Location	Should match
Access Control	OK if different, choose best for intended Storage Class (Archive) use
Owner	Should both be ABS
Retention	OK if different, choose best for intended Storage Class (Archive) use
Volume Set Name	Not set, doesn't matter
Consolidation Criteria	OK if different, choose best for intended Storage Class (Archive) use
Catalog name	OK if different
Number of Streams	Always set to 1 from Conversion utility
Execution Node	Should match
Drive name	OK if different, choose best for intended Storage Class (Archive) use

Only the Administrator at a site can truly determine if two separate Storage Classes (Archives) can be consolidated based upon the intended use of the Storage Class (Archive).

Consolidating Execution Environments

Consolidating Execution Environments is again done by comparing the parameters of pairs of Environments, and then combining those Environments if your decisions indicate they can serve the same purpose.

Use Table D–4 as a guide:

Table D–4 Execution Environment Parameter

Environment Parameter	Matching Criteria
Name	Doesn't Matter choose meaningful name
Data Safety options	OK if different choose best for intended Environment's use
Listing options	OK if different Digital recommends *not* producing listings
Span FileSystems	Should Match
Links Only	Should Match
Original object action	Should Match
User Profile	Will always be ABS from conversion utility, choose PRIVI LEGES best for intended Environment's use
Notification	OK if different choose best for intended Environment's use
Locking options	Should match
Number of drives	OK if different choose best for intended Environment's use
Retry options	OK if different choose best for intended Environment's use
Prolog and Epilog	Should match, or be combined

As with Storage Classes (Archives), only the Administrator at a site can truly determine if two separate Environments can be consolidated based upon the intended use of the Environment.

Implement ABS Policy

This section discusses the steps involved in implementing the ABS Policy as produced by the conversion utility and evaluated by the site Administrator.

Executing the Command Procedures –After you have examined the raw output command files from the conversion utility and done what consolidation or modifications seem appropriate, the command files can simply be executed using the at sign (@) operator at DCL. When each command procedure is invoked, it will:

- Create a Storage Class (Archive)
- Create an Execution Environment
- Create one or more Save Requests and associated schedules.
- Possibly create a Prolog command procedure
- Possibly create an Epilog command procedure

Integrating the Prolog and Epilog Commands –There are several features of an SBK file which are not directly supported by ABS. The conversion utility creates a Prolog command file and an Epilog command file which implement some of these other features.

For example, ABS does not support the Offsite Date or Onsite Date in the SBK file directly. However, by issuing the appropriate MDMS SET VOLUME command, this can be implemented. The conversion utility writes these commands into the Prolog or Epilog command files.

When the conversion utility produces a Prolog and Epilog command procedure, they will be created in the ABS$SLS_CONVERSION directory, and will be called, the same name as the Save Request, but will have "_PROLOG" or "_EPILOG" appended. For example, if you convert the NIGHTLY_SBK.COM, you will end up with the Prolog and Epilog command files:

```
ABS$SLS_CONVERSION:NIGHTLY_ABS_PROLOG.COM
ABS$SLS_CONVERSION:NIGHTLY_ABS_EPILOG.COM
```

If you need the features implemented in the Prolog or Epilog command procedures, you should integrate these into your own Prolog and Epilog command procedures (if any). Both the Execution Environment and the Save Request may have Prolog and Epilog commands associated with them, which are usually the execution of a site specific command procedure. If you want the features implemented in the Prolog or Epilog command procedures produced by the conversion utility, you should invoke them from your site specific command procedure.

SBK Symbols and ABS Logicals

When executing an SBK file, SLS makes various DCL symbols accessible to the prolog and epilog command files you invoke. For example, SLS will define the DCL symbol DO_DISK as the name of the disk being backed up during an SBK execution. The objective is to allow the prolog or epilog commands to produce log messages, or perform other operations based upon the SBK file execution.

ABS provides logical names which provide similar functionality. For example, ABS defines the logical name ABS_OS_OBJECT_SET_1 as the set of files being backed up in the first data movement operation. Thus, it can be used in the place of the FILES_n symbol in an SBK file.

The conversion utility kit provides a command procedure, SLS_SYMBOLS.COM, which attempts to define many of the same DCL symbols as an SBK file does based upon the ABS logical names. For example, it defines the DO_DISK symbol based upon the ABS logical name ABS_OS_OBJECT_SET_1.

See ABS$SLS_CONVERSION:SBK_SYMBOLS.COM command procedure for details on the definition of each SBK symbol. Not all SLS DCL symbols defined are supported by the command procedure.

Disable the SLS SBK Files –It is very important to note that once you have executed the DCL Command procedures produced by the conversion utility, the ABS Save Requests will be executing according to their schedules. This means that you will be doing *both* SLS and ABS backups if you do not disable the SLS SBK files.

The SLS SBK files can be disabled by changing their DAYS_n and TIME_n qualifiers to empty. This causes SLS to no longer schedule the SBK files for execution.

Since SLS and ABS use different media management subsystems, it is highly recommended that you do not use both products on the same node. If you do, you will find that the SLS and MDMS volume databases may get out of synchronization, and there may be contention and other unexpected troubles with drives and jukeboxes. If you wish to stage your SLS to ABS conversion across your network, the following approach is recommended:

Define your database server as your first set of nodes to convert; these nodes will run the MDMS database server

Perform the MDMS conversion on these nodes - see Appendix D.2.

Perform the ABS conversion on these nodes

On other client nodes still running SLS, define the appropriate TAPESTART.COM to point to nodes in the ABS/MDMS database server in the symbol DB_NODES

At this point, your volume, magazine and slot databases are being managed by MDMS, but your client systems are still able to use SLS as the backup paradigm. It is recommended that you convert the remainder of your systems to ABS/MDMS as soon as practical, because some of the more unusual features of SLS/MDMS are not supported by the new MDMS database server.

D.3.1.4 Converting User Backup policy

The conversion utility does not convert User Backup policy automatically. It is only intended to make converting SBK files easier or automatic.

To allow a particular user to do their own backups, follow the steps as outlined in Section . Note that there is no automatic way to set up archives for the entire user population, or a large set of users except by creating a DCL command procedure issuing the correct ABS DCL commands.

D.3.1.5 Monitor ABS Activity

After implementing your backup policy in ABS, you should carefully monitor the activities of ABS until you are confident that your policy is being executed as intended.

There are three ways to monitor ABS activity:

- Show the schedules (MDMS SHOW SCHEDULE, or MDMS SHOW SAVE *).
- Set up Notification criteria on the Environments to send you mail when ABS operations complete. The mail will contain the name of the job and the final status.
- Examine the ABS Log files. All ABS Log files are created in the ABS$LOG: directory, and are called the same name as the Save Request.
- For catalog operations:
- Monitor the Staging Log files
  - These are named ABS$LOG:<catalog_name>_<stream>.LOG
- Monitor the Catalog cleanup log files
  - These are named ABS$LOG:ABS$CATALOG_CLEANUP.LOG

D.3.1.6 Restoring from SLS History Sets

ABS has the ability to restore data backed up by SLS. After you have implemented your backup policy using ABS, it may be necessary to restore data which was backed up using SLS prior to the conversion.

To restore from SLS history sets, you must create an ABS catalog for this purpose. These catalogs are not maintained by ABS and are used only for restore purposes.

ABS catalogs that are created for SLS provide the following features:

- The ability to perform an ABS selective restore operation for data that was previously saved using SLS.
- The ability to perform an ABS lookup operation to search for data objects in an existing SLS history file. The MDMS SHOW CATALOG/FILES command will not work with SLS catalogs.
- The ability to specify wildcard characters, such as an asterisk (*) or percent sign (%), for an ABS operation that searches SLS history sets. You can also specify the ellipsis within square brackets ([...]).

Restrictions:

An ABS catalog created for SLS restore operations imposes the following restrictions:

- You cannot restore Oracle Rdb databases saved using SLS.
- You cannot perform a full restore operation on an image backup used by SLS.
- You cannot perform an incremental restore operation that used SLS.

To create a catalog for restoring data that was saved using SLS, follow these steps:

Step 1. Create a new catalog using the MDMS CREATE CATALOG command.

Step 2. Specify the SLS history set name as the name of the catalog, for example: (name=MY_HIST). The catalog name must match the SLS history set name. For each SLS history set name for which you want to perform a lookup or restore operation, you must create a corresponding ABS catalog.

Step 3. Set the catalog type as SLS.

Step 4. Do not enable staging.

Step 5. If you plan to perform a restore operation, create an Archive which references the new catalog name, specifies a media type, and enables READ only access control. Do not enable WRITE access.

D.3.2 Conversion Utility Reference

D.3.2.1 Command Syntax

```
$ @ABS$SYSTEM:SLS_CONVERT  <wildcard_SBK_spec>  [<match1>] [<match2>…]
```

<wildcard_SBK_spec>

This parameter identifies the set of SBK files to be converted by this command. The string given should *not* include SLS$SYSBAK: or the _SBK.COM suffix. For example, if you want to convert the SBK file SLS$SYSBAK:NIGHTLY*_SBK.COM, you should issue the command:

```
$ @ABS$SYSTEM:SLS_CONVERT NIGHTLY*
```

<match1> … <match7>

These optional parameters allow you to search the SBK files defined by the <wildcarded_SBK_spec> and only process those files which contain ALL of the given strings. The strings must all appear on the same line in the SBK file, since the SLS_CONVERT command procedure uses a /MATCH=AND on the Search command.

D.3.2.2 Output Command File naming and contents

The output of the SLS_CONVERT conversion utility is one DCL command procedure for each SBK file processed. The command procedures will be created in the ABS$SLS_CONVERSION directory.

Each command procedure will be named the same as the SBK file, but substituting “ABS” for “SBK”. For example, if the SBK file SYSTEM_DISK_SBK.COM is converted, the output command procedure will be ABS$SLS_CONVERSION:SYSTEM_DISK_ABS.COM.

Although the command procedures can be executed immediately, it is highly recommended that you review their contents before executing them to ensure that the ABS Policy objects which will be created accurately reflect your intended backup policy.

Each output command file will contain:

- A block of comments indicating that the file was produced by the conversion utility, and the date and time of the conversion.
- The name of the SBK file represented in the command file
- The list of SBK parameters which are not handled by the conversion utility, and the reason they are not.
- An ABS Create Storage command to create a Storage Class (Archive). This corresponds to the MDMS create Archive command.
- An ABS Create Environment command to create an Execution Environment.
- One or more ABS Save commands and ABS Set Save commands to create Save Requests. See Section D.3.1.2 for information on why a single SBK file might produce multiple Save Requests.
- The creation of a Prolog command file. The Prolog command file should be integrated with any site specific prolog command files to complete the functions defined by the SBK.
- The creation of an Epilog command file. The Epilog command file should be integrated with any site specific epilog command files to complete the functions defined by the SBK.

D.3.3 SBK Symbols in ABS Terminology

Table D–5, lists SBK Symbols in ABS Terminology.

Table D–5 SBK Symbols in ABS Terminology

SBK Symbol	ABS Equivalent	Meaning
DAYS_n	Save Request /SCHEDULE and /EXPLICIT	Defines how often the backup oper ations are performed
TIME_n	Save Request /START_TIME	Defines when the backup operation starts
NODE_n	Save Request /SOURCE_NODE	Defines the node in your network where the data resides
BACKUP_TYPE	Save Request /OBJECT_TYPE	Defines the type of data to be backed up or restored.
PRE_PROCESS_FIRST	Environment /PROLOG	Defines a command to be executed when the backup job starts
PRE_PROCESS_EACH	Save Request /PROLOG	Defines a command to be executed prior to each backup operation within a job
POST_PROCESS_EACH	Save Request /EPILOG	Defines a command to be executed when each operation within a job finishes
POST_PROCESS_LAST	Environment /EPILOG	Defines a command to be executed when the backup job completes.

Table D–5 SBK Symbols in ABS Terminology

SBK Symbol	ABS Equivalent	Meaning
NEXT_JOB	Use dependencies in current scheduler interface option if available	Defines what job to run next after the current job completes
SUMMARY_FILE	ABS REPORT SAVE/FULL and ABS Catalogs	Gives overview information about a save operation in a job.
PRIVS	Environment /PRO FILE=(PRIVS)	Defines the set of privileges to be used when executing the operation
FILES_n	Save Request Include Specifi cation	Defines the set of files or other data objects to be backed up or restored.
QUALIFIERS and QUALIFIERS_n	Environment /ACTION, /LOCK and /DATA_SAFETY Save Request /FULL/SINCE=BACKUP /SELECT	Defines characteristics of the save operation, such as the type of data movement, and options for execu tion of the backup.
MNTFLAGS	Not supported. ABS controls mounting of tapes.	Defines how tapes are mounted.
SAVESET_GEN	Not supported. ABS generates the saveset names.	Defines the name of the saveset stored on tape.
PROTECTION	Storage Class (Archive)/ACCESS	Defines what access is available to the backed up data.
MEDIA_TYPE	Storage Class (Archive) /TYPE_OF_MEDIA	Defines which MDMS media type is to be used for backup operations.
DENSITY	Density is an attribute of the MDMS media type object.	Defines the tape density to be used for backup operations.
REEL_SIZE	This maps to the length attribute of the MDMS media type object.	For 9 track tapes, defines the length of the tape (e.g. 2400 feet)
TAPE_POOL	Storage Class(Archive) /TAPE_POOL	Defines the MDMS pool from which tapes are drawn for backup opera- tions.
QUICKLOAD	The MDMS drive attribute "AUTO_REPLY" can be spec- ified on a per-drive basis to determine whether a drive comes on line.	Determines whether MDMS will automatically recognize when a tape drive comes online without opera tor intervention.
QUICKLOAD_RETRIES	Not supported	Defines how long a LOAD request should remain outstanding before being canceled.
PREALLOC	ABS allocates and manages volume sets automatically.	Determines the number of volumes to pre allocate before a backup begins, forming them into a volume set.
AUTOSEL	ABS always automatically selects new volumes to append to volume sets, if needed.	Determines whether SLS is allowed to automatically select new volumes from the volume database if needed.

Table D–5 SBK Symbols in ABS Terminology

SBK Symbol	ABS Equivalent	Meaning
CONTLOADOPT	Logical Name: ABS$DISABLE_SCRATCH_LOADS set to one. By default, ABS will request and accept scratch tapes. The logical can be defined to force specific tapes to be mounted.	Determines whether the operator can substitute a valid tape for the requested tape.
UNATTENDED_BACKUPS	ABS always attempts to per form the backup without oper ator intervention.	Determines whether SYSBAK should default responses to ques tions rather than require operator intervention.
CONTINUE	ABS Storage Class (Archive)Name. Each Storage Class (archive) manages one or more volume sets, and appends data to these volume sets until the Consolidation Criteria is exceeded.	Determines how data is consoli dated onto volume sets.
HISTORY_SET	Catalog Name Storage Class (Archive)/CAT-ALOG	Determines the catalog into which a record of the operations performed and the files backed up is written.
SBUPDT_Q	Not supported. If a catalog supports staging, ABS always performs the catalog update in a detached process.	Determines the Batch Queue in which the System history set update is performed.
SCRATCH_DAYS	Storage Class (Archive) /RETAIN	Determines how long data is saved before the tapes are recycled and cat-alog entries removed.
OFFSITE_DATE ONSITE_DATE	MDMS volumes support OFFSITE_DATE and ONSITE_DATE attributes, and will automatically gener-ate MOVE VOLUME com-mands when these dates are reached.	Determines when the volume sets are moved offsite or onsite (i.e. vaulting).
TAPE_LABELS	Not supported.	Determines if paper labels are printed for the volumes used in the backup
NOTES	The MDMS description field in the volume object is the equivalent attribute.	Store a free form text note in the vol-ume record for the volumes used in the backup.
DRIVE_TYPE	Storage Class (Archive) /DRIVE_LIST	Determines the list of tape drives to be used. It is recommended that MDMS media types be set up cor rectly rather than using this field.
N_DRIVES	Environment /DRIVE_COUNT	Determines the number of tape drives to use during a backup opera tion.

Table D–5 SBK Symbols in ABS Terminology

SBK Symbol	ABS Equivalent	Meaning
PROGRESS	Not supported	Notifies the operator after a certain number of files have been backed up.
REPLY_MSG	Not supported. MDMS issues all OPCOM messages in a standard format	Determines the notification to be performed when each backup opera tion starts and ends.
STATUS_MAIL	Environment /NOTIFY	Determines who should be sent MAIL when the job completes.
LOG_FILE	Not supported. ABS generates a log file in ABS$LOG with the same name as the Save Request.	Determines the name of the log file for the operation.
LISTING_GEN	Environment /LISTING. ABS will generate listing files, but they are always located in ABS$LISTINGS, and are named the same as the Save Request and the operation number.	Determines the name of a backup listing file to be produced from each operation.
FULL	Environment /LIST ING=FULL	Determines if the listing file pro vides all information about backed up files, or only brief information.
PRINT_Q	Not supported.	Determines the Print queue on which the listing file is printed.

D.3.4 ABS Policy Attributes in SBK Terminology

Table D–6, lists ABS Storage Class (Archive) object parameters and their SLS SBK Equivalents.

Table D–6 ABS Storage Class (Archives) and SLS SBK Equivalen

Storage Class (Archive) Parameter	SBK Equivalent	Meaning
Name	CONTINUE	A common name which can be referenced by multiple Save Requests
Primary Archive Loca tion	<None>	For on disk backups, this gives the disk and top level directory where the savesets will be stored.
Primary Archive Type	<None>	This determines whether the Storage Class (Archive) is tape based (type is TAPE) or disk based (type is DISK)
Owner	<None>	Determines the NODE::USER of the owner of the Storage Class (Archive). The owner always has CON TROL access.

Table D–6 ABS Storage Class (Archives) and SLS SBK Equivalen

Storage Class (Archive) Parameter	SBK Equivalent	Meaning
ACL	PROTECTION	Determines the access to the backed up data. ABS provides full ACL based access, while SLS provides only OpenVMS-style system, owner, group and world access.
Tape Pool	TAPE_POOL	The MDMS pool from which volumes will be allocated for backups.
Type of Media	MEDIA_TYPE	The MDMS media type to be allocated for back ups.
Retain	SCRATCH_DAYS	The number of days the backed up data will be saved before the tapes are recycled. Note that a Save Request can specify a retention *shorter or equal to* the value in the Storage Class (Archive).
Consolidation Criteria	CONTINUE	This set of parameters determines how backup savesets will be consolidated onto tapes. For example, if the Consolidation Interval is set to 7 days, savesets will be appended onto a volume set for 7 days before a new volume set is created.
Catalog Name	HISTORY_SET	The name of the catalog which stores data about what files have been backed up, and where they are located.
Number of Streams	<None>	The number of simultaneous Save requests which can be writing into the Storage Class (Archive). Determines the number of MDMS volume sets simultaneously active in the Storage Class (Archive).
Media Location	<None>	The MDMS onsite location field to match for allocat ing volumes for backups.
Drive List	DRIVE_TYPE	The list of specific drives to be used for backup operations in this Storage Class (Archive). Normally, this should be managed through MDMS drive objects.

Table D–7, lists ABS Execution Environment parameters and their SLS SBK Equivalents

Table D–7 ABS Execution Environment Parameter and SLS SBK Equivalent

Environment Parameter	SBK Equivalent	Meaning
Name	<None>	Identifies the Environment for reference by Save Requests
Owner and ACL	<None>	Identifies the owner and access to the Environment.
Data Safety Options	QUALIFIERS	A bitmask containing the data safety options to be applied during the backup. Data safety options include CRC checking, and full data verification.
Listing Options	LISTING_GEN FULL	Determines whether a listing file is produced, and whether the listing is a “full” listing or a “brief” list ing.
Span Filesystem Options	<None>	For UNIX file systems, determines whether the entire filesystem is backed up, even if it crosses multiple physical devices.
Links Only	<None>	For UNIX file systems, determines whether ABS backs up only the logical links, or backs up the data as well.
Compression Options	<None>	For UNIX file systems, determines the type of com pression to be applied to the savesets.
Original Object Action	QUALIFIERS	Determines the action to be taken on the original data objects (e.g. the files backed up). Options include None, Record Backup Date, or Delete
User Profile	PRIVS	Determines the username, privileges and access rights used during the backup operation. The special keyword “<REQUESTER>” indicates the backup operations should be performed with the username, privileges and access rights of the person issuing the ABS Save Command.
Notification Criteria	REPLY_MSG STATUS_MAIL	Determines when notification occurs, what method is used, and who is notified.
Locking Options	QUALIFIERS	Determines how much interlocking is done between the backup in progress and an active file system. Options include Ignore File Writers, and Hot Backup.
Number of Drives	N_DRIVES	Determines the number of tape drives to be used during the backup operations.
Retry Interval and Count	<None>	Determines how often and how many times a failed backup should be retried.
Prolog Command	PRE_PROCESS_FI RST	A command to be executed when the backup starts. Contrast to Save Request Prolog.
Epilog Command	POST_PROCESS_ LAST	A command to be executed when the backup com pletes. Contrast to Save Request Epilog.

Table D–8, lists ABS Save Request parameters and their SLS SBK equivalents.

Table D–8 ABS Save Request Parameter and SLS SBK Equivalent

Save Request Parameter	SBK Equivalent	Meaning
Name	SBK File name	Identifies the group of backup operations to be per formed.
Movement Type	QUALIFIERS	Determines whether the operations are Full, Incre mental or Selective (i.e. individual file) operations.
Source Node	NODE_n	Node on which the data resides.
Include Specification	FILES_n	Identifies the data to be backed up. Multiple include specifications can be given on a single Save Request, and each can have a different Object Type.
Object Type	BACKUP_TYPE	Gives the type of the data. ABS Supports many dif ferent types of data, including OpenVMS Files, UNIX Files, Oracle RDB Databases, and so forth.
Agent Qualifiers	QUALIFIERS	Allows backup agent specific qualifiers to be added to the command used to backup the data.
Since and Before Date	QUALIFIERS	Determine whether data objects to be backed up should be selected based upon creation/modifica tion date.
Exclude Specification	QUALIFIERS	Determines selected data objects to be excluded from the backup.
Storage Class (Archive) Name	<None>	Gives the name of the Storage Class (Archive) into which the data is backed up.
Environment Name	<None>	Gives the name of the Execution Environment to be used for the backup operations.
Start Time	TIME_n	Indicates the time at which the Save Request should start each time it is scheduled. Note that an SBK can provide multiple DAYS_n and TIME_n parameters, an ABS Save Request is restricted to a single Start Time and Interval.
Scheduling Option and Explicit	DAYS_n	Identifies the repeat interval for the Save Request. ABS provides a variety of predefined simple inter vals, such as Daily, Weekly, Monthly, as well as several “complex” intervals, such as Weekly Full with Daily Incremental, and log based schedules. See the information about "Log-n Backup Sched-ules"ABS for a full description of log based sched-ules.
Prolog Command	PRE_PROCESS_E ACH	A command to be executed before each backup operation within the Save Request starts. Contrast to Environment Prolog.
Epilog Command	POST_PROCESS_ EACH	A command to be executed after each backup oper ation within the Save Request completes. Contrast to Environment Epilog.

D.4 Converting ABS V2.x Catalogs to V4.0 Format

If you are upgrading from ABS V2.1, 2.1A or 2.1B, you must convert your catalog format before using them in V4.0.

The catalog upgrade utility upgrades catalogs from their previous format to the new V4.0 format and also deleted expired summary records from those catalogs. A log file named ABS_LOG:ABS_CATALOG_V22_UPGRADE.LOG is generated with information about all of the catalog entries it has modified or deleted.

The catalog upgrade has one update parameter: p1. The input to the p1 parameter is the name of the catalog you wish to upgrade.

You can use the catalog upgrade to upgrade a single ABS catalog or all ABS catalogs. To upgrade an ABS single catalog, specify the catalog name to the p1 parameter. To upgrade all catalogs, enter an asterisk (*) as the wildcard character, or do not supply anything to the p1 parameter.

To run the utility, use the following procedure:

1. Define the following symbol:

```
$ CATALOG_UPGRADE :== $ABS_SYSTEM:ABS$CATALOG_UPGRADE.EXE
```

2. Enter one of the following commands:

```
$ CATALOG_UPGRADE ! upgrades all ABS catalogs

$ CATALOG_UPGRADE * !equivalent to example above

$ CATALOG_UPGRADE ABS_CATALOG ! Upgrades a catalog named ABS_CATALOG
```

Restriction:

The ABS cagtalogs must be quiescent while executing the catalog upgrade utility. The catalog that is being upgraded will be locked; no save, restore, or lookup operations are allowed while the upgrade is occurring.

D.5 Converting ABS V2.x/V3.x RDB Policy Database to ABS V4.0 (MDMS Server Database)

If you are still using an Rdb Policy Database, the command procedure to convert it to RMS was included in previous versions of ABS. You will have to either pull this command procedure from those kits, or contact COMPAQ Customer Support for assistance.

Note

This conversion must be done prior to updating to V4.0.

Conversion of ABS V2.x or 3.x RDB Policy Database is a two step process.

1. Convert the RDB Policy Database to RMS Policy Database using the conversion programs provided with ABS V3.1A or V3.2 before updating to V4.0.

```
$  @ ABS$SYSTEM:ABS$CONVERT_TO_RMS
```

2. Convert the RMS Policy Database to MDMS Database as described in Appendix D.6

D.6 Converting ABS V3.x RMS Policy Database to ABS V4.0 (MDMS Server Database)

Before upgrading to V4.0, do ABS SHOW object_name/FULL of all your objects to an output file, for reference after the upgrade. Then upgrade to V4.0 and follow the steps below to convert the ABS policy database to V4.0.

Before running the conversion ensure that

1. The old ABS Policy database files (ABS$DATABASE:EPCOT.DB%) are not being currently used
2. ABS V4.0 and MDMS V4.0 are configured and running on all the Clients and Servers.

Run the SYS$SYSTEM: ABS$CONVERT_V3_TO_V4.EXE utility to convert the ABS 3.x database to MDMS Database.

The utility exports only the highest version of the policy object. Details of new MDMS objects created are logged in ABS$CONVERT_V3_TO_V4.LOG.

Sample Conversion:

```
$ RUN SYS$SYSTEM:ABS$CONVERT_V3_TO_V4.EXE

Enter the Path of ABS V3.x Policy Database Files [ABS$DATABASE:] :

Converting ARCHIVE Objects...

Converting ENVIRONMENT Objects...

Converting SAVE Objects...

Converting RESTORE Objects...

$
```

Index

A

ABS
    deleting C-1
    installation 3-1
        NT clients 3-4
        OpenVMS clients 3-4
        UNIX clients 3-6
ABS and MDMS Startup Queues 4-3
ABS$SYSTARTUP.COM 4-3
ABS/MDMS Client Software 2-1
ABS/MDMS Graphical User Interface (GUI) Installation 4-8
ABS/MDMS Server Software 2-1
ABS-OMT License B-1
Adding Client Licenses A-1
Adding Windows Parameter 4-7
Allowing ABS Access to All Files on the Windows Systems 4-8

B

Batch Queues 4-3

C

Catalog Conversion D-26
Clients
    NT 3-4
    OpenVMS 3-4
    UNIX 3-6
Command Syntax D-18
Configuration of ABS/MDMS for System Backup to Tape for Oracle Databases 4-11
Configure MDMS 4-4
Configure Remote Tape Drives 4-5
Conversion Overview D-1
Convert ABS Policy Database to MDMS Database 4-5
Convert the MDMS Database D-11
Converting ABS V2.x Catalogs to V4.0 Format D-26
Converting ABS V2.x/V3.x RDB Policy Database to ABS V4.0 (MDMS Server Database) D-26
Converting ABS V3.x RMS Policy Database to ABS V4.0 (MDMS Server Database) D-27
Converting SLS System Backups to ABS D-11
Converting SLS/MDMS V2.X Symbols and Database D-2
Converting User Backup policy D-17
Create a Node Object 4-2

D

Decide Where to Install ABS/MDMS software 2-1
Deleting
    ABS C-1
Determine your use of SLS D-11

G

Granting the Appropriate ABS/MDMS Access Right Identifiers 4-6

H

Hardware
    requirements 2-3
How to Remove ABS/MDMS to Version 4.0 Files 4-11

I

Installation
    ABS 3-1
        NT clients 3-4
        OpenVMS clients 3-4
        UNIX clients 3-6
    hardware requirements 2-3
    optional software 2-4
    privileges 2-2
    process account quotas 2-5
    required processes 2-6
    software requirements 2-4
    system parameters 2-5
    verifying 4-2
Installing ABS for the First Time 4-2
Installing ABS/MDMS 1-1
Introduction D-1
IVP 4-2

J

Java Runtime Environment 4-8

M

MDMS$STARTUP_QUEUE Logical Name 4-3
MDMS$SYSTARTUP.COM 4-3
Meeting OpenVMS Cluster Requirements 4-4
Modifying ABS and MDMS Command Procedures 4-3
Modifying Client Licenses A-1
Monitor ABS Activity D-17

N

Node in MDMS Database 2-8

O

OpenVMS Clients 2-1
Output Command File naming and contents D-18

P

Parameters
    system 2-5
Policy Database D-27
Privileges
    required 2-2
    system 2-2
Process account quotas 2-5
Processes
    required 2-6
Providing Automatic Start Up and Shut Down 4-3

Q

Quotas
    process account 2-5

R

RDF 4-5
RDF Configuration 2-8
Removing Client Licenses A-3
Removing SLS/MDMS V2.x Automatic Startup 4-4
Requirements
    hardware 2-3
    optional software 2-4
    privileges 2-2
    process account quotas 2-5
    processes 2-6
    software 2-4
    system parameters 2-5
Restoring from SLS History Sets D-17
Right Identifiers 4-6
Running the GUI 4-10

S

Showing Client Licenses A-2
SLS V2.x to ABS V4.0 Conversion Process D-11
Software
    requirements 2-4
Software requirements
    optional 2-4

T

TCPIP Quotas for Clients 4-7
Tru64 UNIX C lient 3-6
Tru64 UNIX Clients 2-2

U

UNIX 3-6
Updating ABS/MDMS 1-1
upgrade from ABS-OMT B-1
Upgrading from SLS to ABS/MDMS 1-1

V

Verification
    installation 4-2
Verifying ABS Installation 4-2
Verifying Windows and Tru64 UNIX Client Quotas 4-7

W

Windows Client 3-4
Windows Clients 2-2
Windows Parameter 4-7